

Joni Liukkonen
SÄHKÖMOOTTOREIDEN KOEKÄYTTÖJÄRJESTELMÄ

Kone- ja tuotantotekniikan koulutusohjelma
2014

SÄHKÖMOOTTOREIDEN KOEKÄYTTÖJÄRJESTELMÄ

Liukkonen, Joni
Satakunnan ammattikorkeakoulu
Kone- ja tuotantotekniikan koulutusohjelma
Huhtikuu 2014
Ohjaaja: Suvela, Timo
Sivumäärä: 48
Liitteitä: 8

Asiasanat: Profinet, Taajuusmuuttajat, Käyttöliittymät, ABB

Tämän opinnäytetyön tarkoituksena oli suunnitella, testata ja käyttöönottaa rakennettavan sähkömoottoreiden koekäyttöjärjestelmän automaation ohjaus. Työn toimeksiantajana oli Sataservice Oy ja asiakkaana eräs teollisuusyritys. Työn päätavoitteet olivat logiikkaohjelman toteutus ja käyttöliittymän kehittäminen kosketusnäytölle. Ohjelmointityön lisäksi työhön kuuluivat järjestelmän testaus toimeksiantajan tiloissa ja käyttöönotto asiakasyrityksen tiloissa. Työssä perehdytään Profinet-kenttäväylän toimintaperiaatteeseen, taajuusmuuttajan kontrolloimiseen kenttäväylän kautta sekä käsitellään projektin kulun eri vaiheet ja kuvaillaan saadut lopputulokset.

Projekti aloitettiin tutustumalla rakennettavaan järjestelmään ja suunnittelemalla käyttöliittymän ja logiikkaohjelman luonnokset. Työn alkuperäinen aikataulu oli 8 viikkoa, joka viivästyi kuitenkin monilla viikoilla laitetoimitusten takia. Tämän johdosta aikaa jäi teoriaan perehtymiseen ja itselle uusien kehitystyökalujen opiskeluun.

Seuraavassa vaiheessa ohjelmaa ja käyttöliittymää testattiin rakentamalla väliaikainen kenttäväylä logiikan, taajuusmuuttajan ja kosketusnäytön välille. Viimeisessä työvaiheessa ohjelmaa ja käyttöliittymää testattiin asiakkaan tiloissa.

Lopputuloksena saatiin asiakkaalle asennetun sähkömoottoreiden koekäyttöjärjestelmän logiikkaohjelma ja käyttöliittymä. Lisäksi järjestelmä saatiin käyttöönotettua asiakkaan tiloissa viimeistä moottoritestä lukuun ottamatta.

TESTING SYSTEM FOR ELECTRIC MOTORS

Liukkonen Joni

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Mechanical and Production Engineering

April 2014

Supervisor: Suvela, Timo

Number of pages: 48

Appendices: 8

Keywords: Profinet, Frequency converters, HMI, ABB

The purpose of this thesis was to plan, develop and test an automation control of the electric motor testing system. The client of this thesis was Sataservice Oy and the end-customer of the final system was an industrial company from Western Finland. The main goal was to plan and produce a logic program and a user interface for testing system. In addition to programming process, other tasks were to test the system at the client's workshop and start-up the system at the customer's premises. This report focuses on the theory of Profinet-technology, how frequency converter is controlled via fieldbus and describes different phases of the project. The report also introduces final results of the project.

The project started by exploring the structure of the system. Also the draft of the user interface and the program logic was designed. Original project schedule was 8 weeks, which was postponed for several weeks because of the delayed equipment deliveries. On the other hand, these delays gave time to peruse Profinet-technology.

In the next phase, the logic program and the user interface were tested with temporarily built-in fieldbus. The temporary fieldbus was built between programmable logic controller, frequency converter and touchscreen. At the final stage, the program and the user interface were tested at the customer's premises.

As a result, this project produced a logic program and a user interface for electric motor testing system. Additionally, the implementation of the new system completed successful with the exception of final test with real test motors.

SISÄLLYS

1	JOHDANTO	6
1.1	Sataservice Oy	6
1.2	Projektin esittely	6
2	PROFINET-KENTTÄVÄYLÄ	8
2.1	Kommunikointitavat	9
2.2	Kommunikointiluokat	9
2.3	Datakehys	11
2.4	Reaaliaikaisuuden toteuttaminen	12
2.5	Laitteet	14
2.6	Laitteiden identifiointi	15
3	TAAJUUSMUUTTAJA	16
3.1	Moottorin ohjaus	16
3.2	Taajuusmuuttajan konfigurointi ja ohjelmointi	17
3.2.1	Logiikan ja taajuusmuuttajan rajapinta	17
3.2.2	Ohjaus- ja tilasanat	19
3.2.3	Parametrit	19
4	AUTOMAATIOPROJEKTIN ELINKAARIVAIHEET	19
5	AUTOMAATION OHJAUKSEN SUUNNITTELU	21
5.1	Vaatimusten määrittely	21
5.2	Laitteistovalinnat	22
6	OHJELMOINTI	23
6.1	Laitteistomäärittely	23
6.2	Profinet-väylän käyttöönotto ja konfigurointi	24
6.3	Parametrien määrittäminen taajuusmuuttajaan	27
6.4	Logiikkaohjelma	28
6.4.1	Ohjelmistoarkkitehtuuri	28
6.4.2	Pääohjelma	28
6.4.3	Ohjaus- ja tilasanojen käsittely	29
6.4.4	Taajuusmuuttajan ohjaus	31
6.4.5	Parametrien käsittely	32
6.5	Käyttöliittymän kehittäminen	33
6.5.1	Käyttöliittymän rakenne	33
6.5.2	Pääsivu	34
6.5.3	Käyttökohtaiset sivut	35
6.5.4	Huoltosivu	37
6.5.5	Käyttöliittymän integrointi logiikkaan	37

7 KÄYTTÖÖNOTTO	39
7.1 Järjestelmän testaus	39
7.2 Käyttöönotto asiakkaan tiloissa	41
8 POHDINTA	43
8.1 Yhteenveto	43
8.2 Kehitysideoita	45
8.3 Oppimistulokset	45
LÄHTEET	47
LIITTEET	

1 JOHDANTO

1.1 Sataservice Oy

Sataservice Oy on Länsi- ja Etelä-Suomessa toimiva kunnossapitoalan yritys. Perinteisten kunnossapitotöiden lisäksi Sataservice Oy tarjoaa muita teollisuuden sähkö-, automaatio-, ja mekaniikkapalveluita aina raskaan kaluston huolloista automaatiojärjestelmien ohjelmointiin. (Sataservice Oy:n www-sivut 2012.)

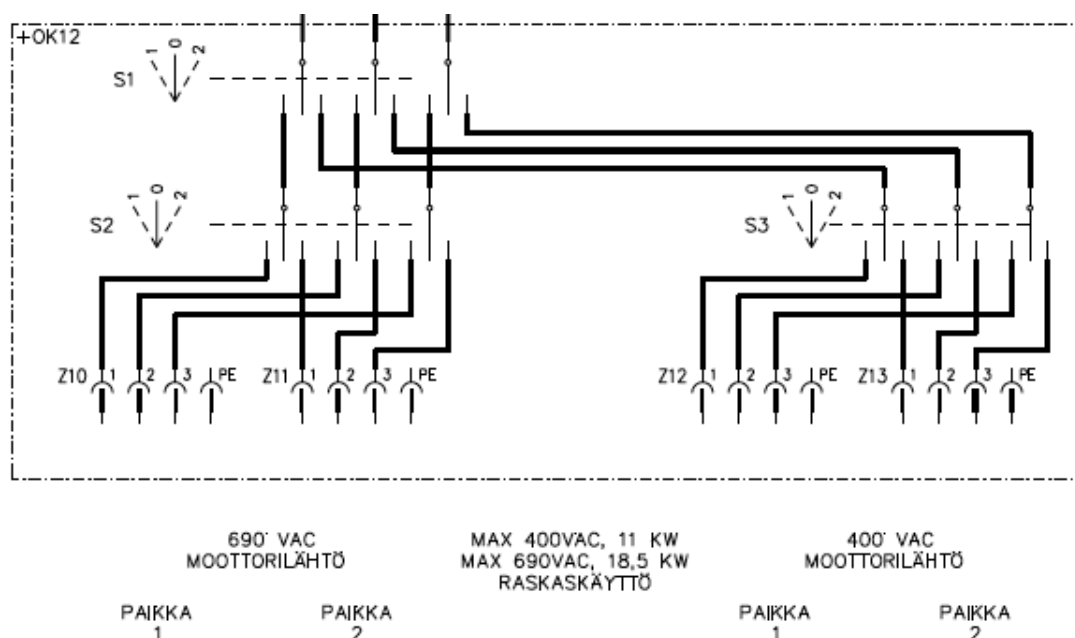
Sataservice Oy perustettiin vuonna 2003 kahden henkilön toimesta. Nykyään Sataservice konserniin kuuluvat myös Sataservice Food Oy, Kolmikoneistus Oy ja Rauman Sähkökonehuolto Oy. Sataservice Oy:n toimipisteitä on jo monissa eri Länsi- ja Etelä-Suomen kaupungeissa, joista Raumalla pääkonttorissa keskitytään konsernin johtoon sekä erilaisten automaatioprojektien suunnitteluun ja käyttöönottoon avaimet käteen -periaatteella. Samoissa tiloissa toimivat myös Kolmikoneistus Oy ja Rauman Sähkökonehuolto Oy. Sataservice Oy:n palveluksessa on tällä hetkellä noin 300 henkilöä. (Sataservice Oy:n www-sivut 2012.)

1.2 Projektin esittely

Tämän opinnäytetyön tavoitteena on toteuttaa asiakasyritykselle rakennettavan sähkömoottoreiden koekäyttöjärjestelmän automaation ohjaus. Työn päätavoitteet ovat logiikkaohjelman toteutus ja käyttöliittymän kehittäminen kosketusnäytölle. Ohjelmointityön lisäksi työhön kuuluvat järjestelmän testaus toimeksiantajan tiloissa ja käyttöönotto asiakasyrityksen tiloissa.

Rakennettavan järjestelmän automaatio koostuu kolmesta taajuusmuuttajasta, ohjelmoitavasta logiikasta ja kosketusnäytöstä. Laitteiden tiedonvaihto toteutetaan Profinet-verkon avulla. Järjestelmän tehtävä on koekäyttää teholtaan eri suuruisia sähkömoottoreita. Sähkömoottoreita koeajetaan täydellä nopeudella, kolmella eri taajuusmuuttajakäytöllä. Jokaiseen taajuusmuuttajakäyttöön on mahdollisuus valita yksi kahdesta moottorilähdöstä, joissa toisessa on mahdollisuus koeajaa 400 VAC moot-

toreita ja toisessa 690 VAC moottoreita. Lisäksi taajuusmuuttajakäytöissä 1 ja 2 on mahdollisuus valita käytettävän moottorilähdön lisäksi koestettavalle moottorille käyttöpaikka. Esimerkiksi taajuusmuuttajan käyttämä moottorilähtö valitaan kytkimellä S1 ja käyttöpaikka kytkimillä S2 tai S3 (Kuva 1).



Kuva 1. Käyttö 1, koeajettavan laitteen moottorilähdön ja käyttöpaikan valinta tehdään keskuksessa, johon moottori liitetään pistotulpalla.

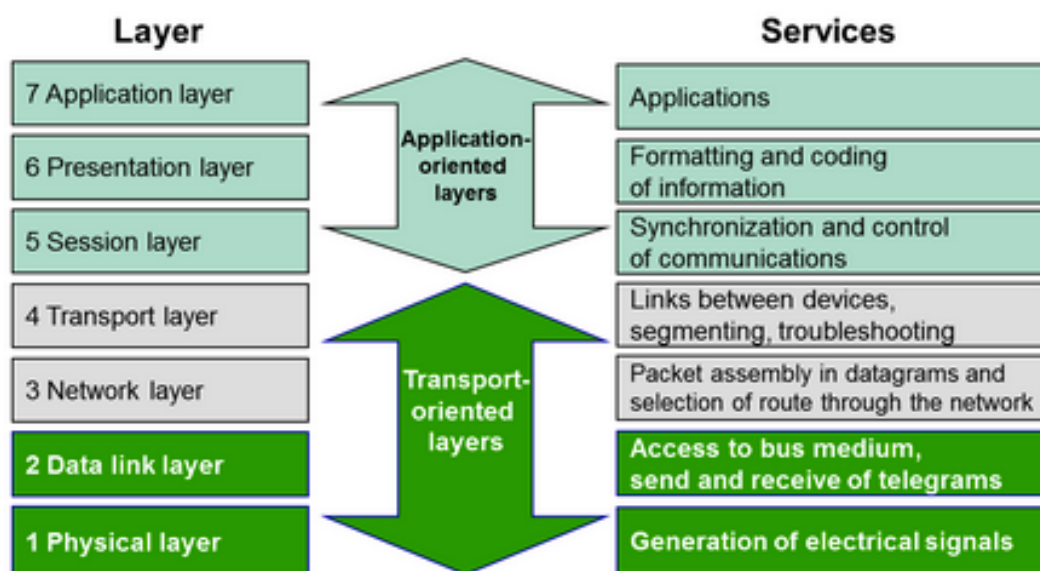
Koekäytettävät moottorit ovat erikokoisia, joten moottorin jännite, virta ja muut moottorin tiedot vaihtelevat moottorin koon mukaan. Taajuusmuuttaja ohjaa moottoria näiden tietojen avulla, joten jokaisen moottorinvaihdon yhteydessä taajuusmuuttajiin tulee vaihtaa käytettävää moottoria vastaavat moottoriparametrit. Parametrien vaihtamisen nopeuttamiseksi ja helpottamiseksi käytetään kosketusnäyttöä. Näytöltä valitaan käytettävän moottorin kilpiarvoja vastaavat arvot ja siirretään ne moottoreita ohjaaviin taajuusmuuttajiin. Tämän lisäksi kosketusnäytöllä näytetään käyttökohtaisesti hetkelliset oloarvot ja vikatiedot. Jokainen moottori käynnistetään omasta Start-painikkeesta ja kaikki moottorit pysäytetään yhteisestä Stop-painikkeesta. Painikkeet sijoitetaan erilliseen käsiohjaimeen.

Työ rajataan siten, että sähkökeskusten kokoaminen, laitteiston asennus ja kaapeleiden vedot tehdään aikataulusyistä ja työn laajuuden takia asentajien toimesta.

2 PROFINET-KENTTÄVÄYLÄ

Kenttäväylä on teollisuudessa käytettävä digitaalinen tiedonsiirtoteknologia automaatiolaitteille. Kenttäväylän avulla ohjaavat laitteet, kuten ohjelmoitava logiikka voivat kommunikoida kenttälaitteiden kanssa. Tässä projektissa käytettävä Profinet-kenttäväylä on PROFIBUS & PROFINET Internationalin kehittämä Ethernet-protokolla perustuva tekniikka. Profinet on teollisuuskäyttöön sopivilla ominaisuuksilla muunneltu versio tavallisesta Ethernet-protokollasta. Profinet-tekniikka mahdollistaa muun muassa aikakriittisen ja ei-aikakriittisen tiedonsiirron laitteiden välillä. (PI North America 2014.)

Internetissä tapahtuvaa tiedonsiirtoa kuvataan usein 7-kerroksisen OSI-mallin avulla (Kuva 2). OSI-mallissa siirrettävän datan on kuljettava jokaisen kerroksen läpi, ennen kuin se voidaan vastaanottaa päätelaitteella tai lähettää verkon yli halutulle laitteelle. Jokaisen kerroksen tehtävä on vastaanotettaessa purkaa ja lähetettäessä pakata data seuraavassa kerroksessa tehtäviä toimenpiteitä varten. (GE Intelligent Platforms 2013.)



Kuva 2. OSI-malli (GE Intelligent Platforms 2013)

Kuten Ethernetin tiedonsiirrossa, myös Profinet-tekniikassa hyödynnetään OSI-mallin kerroksia 1-4 ja 7. Kerroksissa 1 ja 2 toimivat IEEE 802.3-standardin määrittelemä Ethernet-protokolla. Kerros 3 on IP- ja kerros 4 TCP-protokollan käytössä. 7. kerros on käyttäjän näkemien sovellusten käytössä. (GE Intelligent Platforms 2013.)

Jokaisella kerroksella on oma tehtävänsä tiedonsiirrossa:

- **7. Sovelluskerros** (Application layer), jossa luodaan lähetettävä ja näytetään vastaanotettava tieto käyttäjälle todellisen ohjelman avulla
- **4. Kuljetuskerros** (Transport layer) luo yhteyden kahden laitteen välille ja siirtää datapaketit kokonaisuudessaan
- **3. Verkkokerros** (Network layer) vastaa datapakettien reitittämisestä oikeaan osoitteeseen ja huolehtii mm. virheen korjauksesta ja ruuhkan hallinnasta
- **2. Siirtokerros** (Data link layer), jossa koodataan lähtevä data biteiksi ja puretaan tuleva data biteistä
- **1. Fyysinen kerros** (Physical layer) siirtää tiedon kaapelia pitkin binäärikoodauksen avulla

(GE Intelligent Platforms 2013)

2.1 Kommunikointitavat

Profinet mahdollistaa syklisen ja asyklisen kommunikoinnin laitteiden välillä. Syklisessä kommunikoinnissa väylässä siirrettäviä sanomia päivitetään jatkuvasti. Syklisen tiedonsiirto on lähes reaaliaikaista. Syklisen kommunikoinnin lisäksi Profinet-tekniikan avulla voidaan toteuttaa asyklista eli tapahtumapohjaista kommunikointia laitteiden välillä. Tapahtumapohjaiset sanomat lähetetään vain silloin, kun käyttäjä niin haluaa. Asyklisessä kommunikoinnissa voi kuitenkin esiintyä viiveitä, joten sitä ei suositella käytettävän aika-kriittisissä kohteissa. (ABB Industrial drives 2011.)

2.2 Kommunikointiluokat

Profinet-väylässä tiedonsiirto luokitellaan reaaliaikaisuuden ja tiedonsiirtotavan mukaan eri kategorioihin:

- NON-REAL TIME (TCP/IP, asyklinen tiedonsiirto)
- REAL TIME 1 (RT, syklinen tiedonsiirto)
- REAL TIME 2 (IRT, syklinen tiedonsiirto)
- REAL TIME 3 (IRT, syklinen tiedonsiirto)

(Control Builder 2013)

NRT (Non Real Time) eli ei-aikakriittistä tiedonsiirtokanavaa käytetään, kun tiedonsiirtoon kuluva aika ei ole laitteen toiminnan kannalta kriittinen tekijä. NRT-sanomat ovat asyklisia eli tapahtumapohjaista liikennettä ja ne lähetetään normaalin OSI-mallin mukaan TCP- ja IP-kerroksien kautta. Ei-aikakriittisiä toimenpiteitä ovat esimerkiksi Profinet-laitteen parametrien konfigurointi, diagnostiikka ja järjestelmän käynnistäminen, jolloin TCP/IP-protokollan suuremmalla viiveellä ei ole suurta merkitystä. Viive tässä luokassa on noin 100 ms. (Gory 2005.)

RT 1 (Real Time) eli reaaliaikaista kanavaa hyödynnetään esimerkiksi kappaletavara-automaatiossa, jossa tarvitaan nopeaa reaaliaikaista tiedonsiirtonopeutta. RT-luokassa sanomat lähetetään suoraan Profinet-laitteelle, jolloin ne ohittavat TCP- ja IP-kerrokset. Tämä poistaa TCP- ja IP-kerroksissa tapahtuvan sanomien tiedon valmisteluun ja käsittelyyn kuluvan ajan. Tiedonsiirto tässä luokassa on pääasiassa syklistä, mutta esimerkiksi hälytys- ja viestisanomia voidaan lähettää asyklisesti. Viive tässä luokassa on noin 10 ms. (Gory 2005.)

RT 2 ja RT 3 kuuluvat IRT (Isochronous Real Time) tiedonsiirtoluokkaan, joka mahdollistaa reaaliaikaisen erittäin nopean tiedonvaihdon IO-ohjaimen ja IO-laitteen välillä. IRT-kanavaa voidaan käyttää mm. liikkeenohjauksen sovelluksissa. Viive tässä luokassa on noin 1 ms. (Gory 2005.)

NRT- ja RT 1 -kommunikointiluokkia on mahdollista käyttää tavallisilla Ethernet-kytkimillä rakennetussa Profinet-verkossa. Käytettävien kytkinten on kuitenkin tuettava VLAN-kehyksien priorisointia, jotta RT 1 luokan lupaama nopeus on mahdollista toteuttaa. Sen sijaan kommunikointiluokkien RT 2 ja RT 3 käyttö vaatii erityisen reaaliaikaista tiedonsiirtoa tukevan kytkimen. Alle 1 millisekunnin viiveellä toimiva järjestelmä voidaan toteuttaa esimerkiksi Siemensin kehittämän ERTEC-tekniikan

(Enhanced Real-Time Ethernet Controller) avulla. ERTEC-ohjain sisältää sisäänrakennetun kytkimen ja 32-bittisen ARM-mikroprosessorin. Ohjaimeen integroidun mikroprosessorin käyttäminen mahdollistaa lähes viiveettömän tiedonsiirron laitteiden välillä, koska kaiken reaaliaikaisen tiedon käsittely tapahtuu ERTEC-ohjaimessa. Tämä vähentää myös muiden laitteiden mikroprosessoreiden kuormaa huomattavasti, joka osaltaan mahdollistaa nopeimman mahdollisen tiedonsiirron laitteiden välillä. IRT-kommunikointiluokan reaaliaikaisuus voidaan toteuttaa myös pelkällä IRT-tiedonsiirtoa tukevalla kytkimellä. (Siemens AG 2008.)

2.3 Datakehys

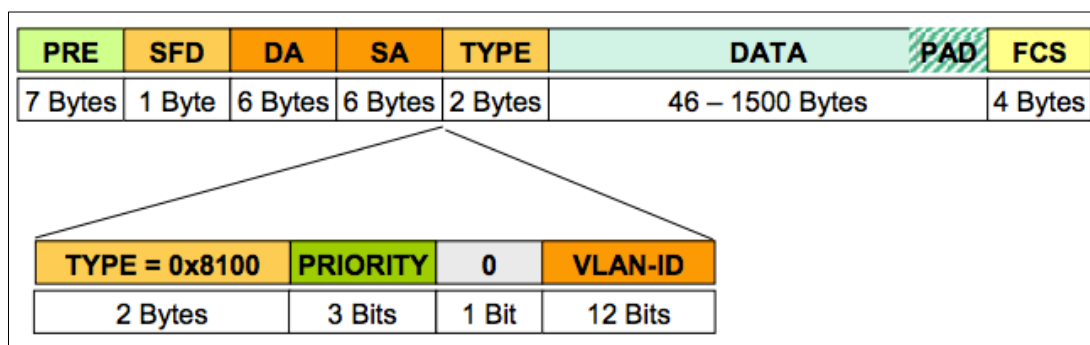
Tiedonsiirrossa käytettävät kehykset ovat eräänlaisia biteistä koostuvia datapaketteja, jotka toimitetaan osoitteesta A osoitteeseen B. Profinet IO perustuu Ethernet-protokollaan, joten tiedonsiirrossa hyödynnetään tavallista Ethernet-kehystä. (Jaakonhuhta 2005, 83.)

Standardin IEEE 802.3 mukainen Ethernet-kehys (Kuva 3) koostuu seuraavista osista:

- **Preamble** (Alkutahdistus). Alkutahdistus virittää vastaanottavan osapuolen.
- **SFD** (Kehyksen alkuerote). Alkuerote erottaa kehyksen alun oikeasta sisällöstä.
- **Destination Address** (Kohdeosoite). Ethernet-kehyn määränpään MAC-osoite.
- **Source Address** (Lähdeosoite). Ethernet-kehyn lähteen MAC-osoite.
- **VLAN-kehys**
 - **Type** (Protokollatunniste). Kertoo VLAN-kehyn tyypin.
 - **Priority** (Käyttäjän prioriteetti). Prioriteetti mahdollistaa 8:n eri palveluluokan muodostamisen aikakriittiselle datalle.
 - **VLAN-ID** (Tunniste). Tunniste kertoo mihin VLAN-verkkoon sanoma kuuluu.
- **Type**. Ethertype kertoo kehyksen tyypin ja pituuden.
- **Data** (Informaatio). Sisältää siirrettävän tiedon.

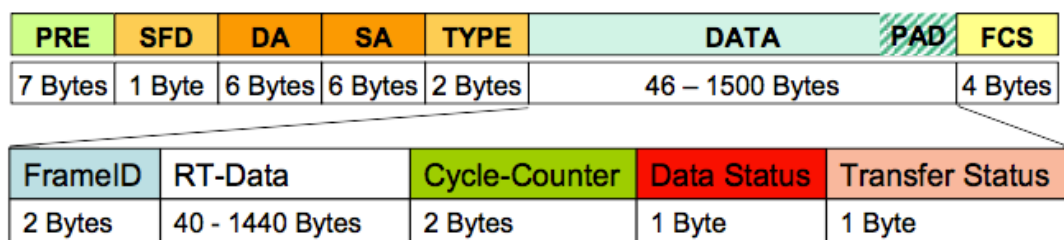
- **PAD** (Täyte). Alle 64-bittisiin kehyksiin jäänyt tyhjä tila täydennetään täyteellä.
- **FCS** (Tarkistussumma). Vastaanottajan laskema 4 oktetia pitkä tarkistussumma, jonka tulee täsmätä lähettäjän laskemaan summaan.

(Jaakonhuhta 2005, 86)



Kuva 3. VLAN-tunnisteen sisältävän Ethernet-kehiksen rakenne (IFAC 2005)

Ethernet-kehiksen datakentässä lähetetään varsinainen Profinet-kehys. Profinet-kehys koostuu viidestä eri tietokentästä, jotka ovat FrameID, RT-Data, Cycle-Counter, Data Status ja Transfer Status (Kuva 4). FrameID määrittelee Profinet-kehiksen tyyppiä. RT-Data kentässä siirretään varsinainen data, jota voi olla esimerkiksi taajuusmuuttajan ohjaus- ja tilasanat sekä parametrien muutokset valitun PPO-tyypin mukaan. Cycle-Counter, Data Status ja Transfer Status kertoo tietoja kulu-
neesta ajasta, datasiirrosta ja sen onnistumisesta. (IFAC 2005.)

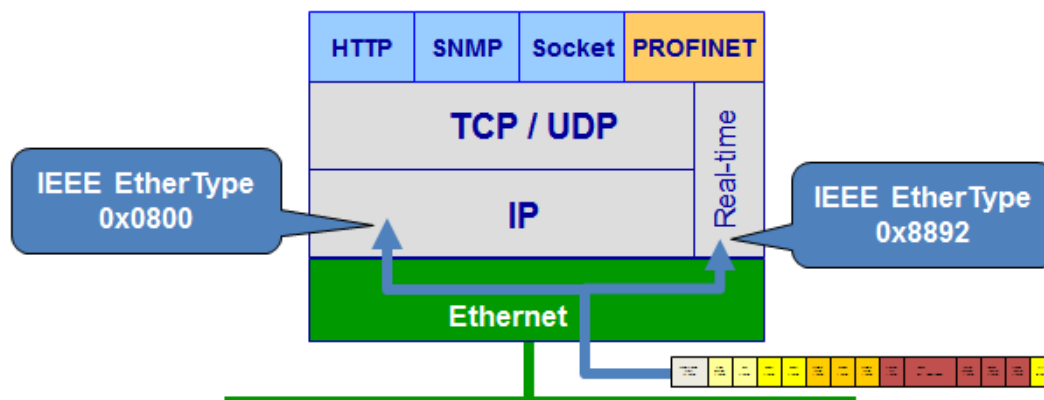


Kuva 4. Profinet 0x8892-kehys (IFAC 2005)

2.4 Reaaliaikaisuuden toteuttaminen

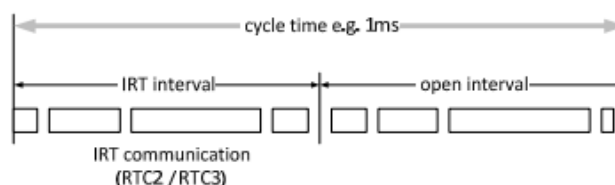
RT 1 -tiedonsiirtoluokassa reaaliaikaisuus toteutetaan valvomalla tiedonsiirtokehiksen tyyppiä. Kehiksen tyyppi määrittelee sen, miten kehys ohjataan vastaanottajalle

(Kuva 5). Esimerkiksi saapuva Profinet -reaaliaikakehys on tyyppiä 0x8892, joten se ohjataan suoraan Profinet-sovellukselle. Tämä vähentää Profinet-sanomien käsittelyyn kuluva aikaa ja parantaa reaaliaikaisuutta, koska merkittävä osa tästä ajasta kuluu tiedon prosessointiin TCP/IP-kerroksissa. (GE Intelligent Platforms 2013.)



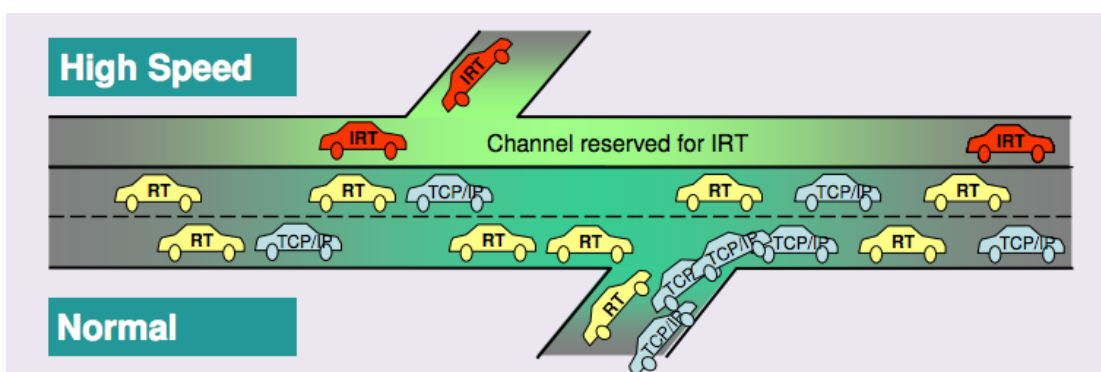
Kuva 5. Profinet-sanomien reaaliaikaisuuden toteuttaminen ja sanomien ohjaaminen OSI-mallissa (GE Intelligent Platforms 2013)

Reaaliaikasanomien lisäksi samalla kanavalla kulkevat TCP/IP-sanomat. Tämän johdosta kehystyyppien tunnistamisen lisäksi tarvitaan tiedonsiirron priorisointia, jotta reaaliaikasanomat säilyttävät lyhyen vaste-ajan. Ethernet-kehysten priorisointi toteutetaan lisäämällä Ethernet-kehykseen aiemmin kuvattu VLAN-tunniste. VLAN-tunniste määrittelee kohdeverkon ja kertoo siirrettävän sanoman prioriteetin. Ethernet-kytkin tunnistaa kehystyyppin ja prioriteetin mukaan siirtää kehysten jonossa alemman prioriteetin kehysten eteen. (Jaakonhuhta 2005, 88.)



Kuva 6. IRT-tiedonsiirron ja vapaan tiedonsiirron kaksi vaihetta (Bernhard, H. & Mottok, J. 2014, 2)

RT 2 ja RT 3-tiedonsiirtoluokissa erittäin nopea IRT-tiedonvaihto saavutetaan toteamalla ensin Ethernet-kehysten tyyppi, jonka jälkeen kehys tunnistetaan IRT-kehykseksi Data-kentästä löytyvällä FrameID-tiedolla. Väylän päivitys on vaiheistettu kahteen eri osioon (Kuva 6). Toinen osio on varattu IRT-kehyksille, jotka siirretään omalla kanavalla, erillään RT- ja TCP/IP-sanomista (Kuva 7). Toinen osio puolestaan varataan tavalliselle RT- ja TCP/IP-liikenteelle. Näin voidaan taata IRT-sanomien reaaliaikaisuus ja mahdollistaa täysin esteetön ja nopea tiedonvaihto IO-laitteen ja IO-ohjaimen välillä. Tämä nopeusluokka vaatii IRT-tuen Profinet-verkon kytkimiltä. (Suvela 2011.)



Kuva 7. IRT-kanavan tieto siirretään erillään muusta liikenteestä (Gory 2005)

2.5 Laitteet

Profinet-väylässä käytettävät laitteet jaetaan kolmeen eri luokkaan.

- **IO-ohjain** (IO-controller) on lähes aina ohjelmoitava logiikka, joka sisältää automaatiota ohjaavan ohjelman. IO-ohjain kommunikoi väylään määriteltujen IO-laitteiden (IO-device) kanssa ja kontrolloi niitä. Tiedonvaihto tapahtuu logiikan prosessikuvan kautta.
- **IO-laite** (IO-device) on väylän avulla ohjaimen liitetty kenttälaite. IO-laite lähettää ohjaimelle tietoa omasta tilastaan ja voi esimerkiksi ottaa vastaan käskyjä ohjaimelta. Tässä projektissa IO-laitteita ovat kolme taajuusmuutta-

jaa, jotka on liitetty väylään Ethernet-adapterilla. Muita IO-laitteita voivat olla esimerkiksi hajautettu IO ja erilaiset anturit.

- **IO-supervisor** on tietokoneen ohjelmointityökalu tai järjestelmän käyttöliittymä (HMI), jota käytetään huolto- ja kunnossapitotehtäviin, käyttöönottoon sekä järjestelmän ohjaamiseen IO-ohjaimen rinnalla.

(PI North America 2011.)

2.6 Laitteiden identifiointi

Profinet IO-väylässä jokaiselle laitteelle on annettava symbolinen nimi (Device name). Tämän nimen avulla jokainen laite voidaan yksilöllisesti tunnistaa Profinet-järjestelmässä. IO-ohjain käyttää symbolista nimeä IP-osoitteen antamiseen laitteille. IP-osoitteet määrätään laitteille DCP-protokollaa käyttäen, joka on usein valmiiksi integroitu jokaiseen IO-laitteeseen. (PI North America 2011.)

Lisäksi jokaiselle laitteella on MAC-osoite, joka on jokaiselle laitteelle yksilöllinen. MAC-osoite ei ole dynaaminen niin kuin IP-osoite, eli MAC-osoite pysyy samana ja onkin usein jo valmiiksi koodattu komponenttiin. MAC-osoitetta on kuitenkin mahdollista muuttaa ohjelmallisesti jälkeinpäin. MAC-osoite koostuu kuudesta kahden heksaluvun parista. Kolme ensimmäistä heksalukuparia, eli bitit 47-24, ovat varattu OUI-tunnistetta (Organizationally Unique identifier) varten. Profinet International -organisaation OUI-tunniste on 00-0E-CF ja se muodostetaan MAC-osoitteeseen kuvan 8 mukaisesti. Loppuosa MAC-osoitteesta (bitit 0-23) muodostavat komponenttikohtaisen heksalukujonon. (PI North America 2011.)

Bit significance 47 to 24						Bit significance 23 to 0					
0	0	0	E	C	F	X	X	X	X	X	X
Company code → OUI						Consecutive number					

Kuva 8. MAC-osoitteen rakenne (PI North America 2011)

3 TAAJUUSMUUTTAJA

Taajuusmuuttaja on laite, jota käytetään muuttamaan sähköverkosta saatavan jännitteen itseisarvoa ja taajuutta. Taajuusmuuttajia käytetään sähkömoottoreiden ohjaamiseen ja säätämiseen teollisuudessa. (ABB Group 2014.)

3.1 Moottorin ohjaus

ABB:n ACS880-taajuusmuuttajalla voidaan ajaa sähkömoottoria kahdella eri moottorinsäätötavalla. Suora momenttisäätö eli DTC-säätötapa säätää sähkömoottorin roottorin nopeutta ja vääntömomenttia. Tässä säätötavassa taajuusmuuttaja säätää moottorin nopeuden vakioiksi, jolloin tätä varten tarvitaan tarkka malli moottorista. (ABB Industrial drives 2013.)

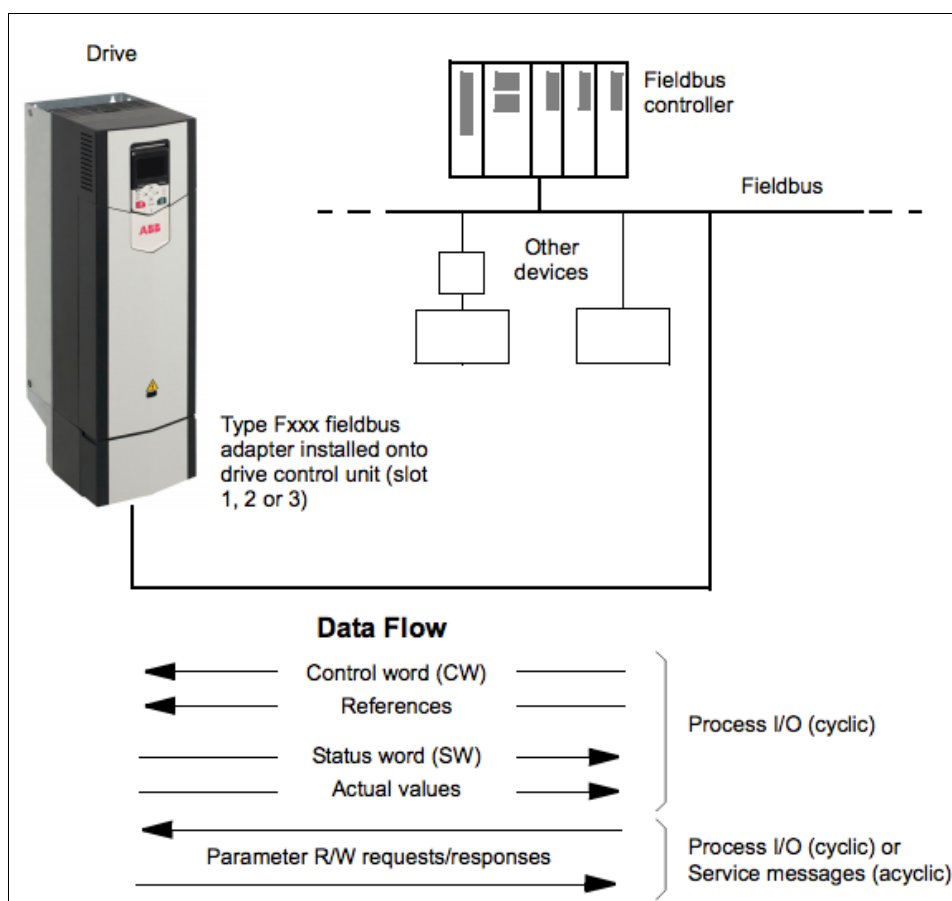
Moottorimalli käytettävästä moottorista saadaan tekemällä moottorille tunnistusajo. Tunnistusajo voidaan tehdä joko ID-ajolla tai tunnistusmagnetoinnilla. ID-ajossa taajuusmuuttaja pyörittää kuormittamatonta moottoria noin 90 sekuntia 50-100 % nopeudella nimellisnopeudesta ja luo muistiin tarkan moottorimallin. Tunnistusmagnetoinnissa taajuusmuuttaja magnetoi moottorin tasavirralla, jolloin AC-oikosulkumoottorin akseli ei pyöri. Tunnistusmagnetointi luo epätarkemman moottorimallin kuin ID-ajo. (ABB Industrial drives 2013.)

Skalaarisäätö puolestaan säätää staattorin taajuutta ja jännitettä nimellistaajuuden ja -jännitteen mukaan. Toisin kuin DTC-säätötapa, skalaarisäätö ei vaadi käytettävän moottorin tarkkaa moottorimallia (ABB Industrial drives 2013). Tässä projektissa säätötavaksi valittiin skalaariohjaus, koska koeajojärjestelmässä moottoreita testataan vakionopeudella. Skalaariohjaus ei ole niin tarkka kuin DTC-säätö, mutta soveltuu hyvin tähän käyttötarkoitukseen, koska moottorin tunnistus- tai magnetointiajoa ei vaadita.

3.2 Taajuusmuuttajan konfigurointi ja ohjelmointi

3.2.1 Logiikan ja taajuusmuuttajan rajapinta

Oletuksena ABB:n ohjelmistossa käytettävä RT 1 -tiedonsiirtoluokka käyttää tiedonvaihdossa syklistä kommunikointia. Syklisessä tiedonvaihdossa väylään lähetettäviä tietoja päivitetään jatkuvasti tasaisin väliajoin. Tiedonvaihto PLC:n ja taajuusmuuttajan välillä koostuu ohjaus- ja tilasanoista, ohjearvoista, oloarvoista sekä parametri-kyselyistä (Kuva 9). (ABB Industrial drives 2011.)



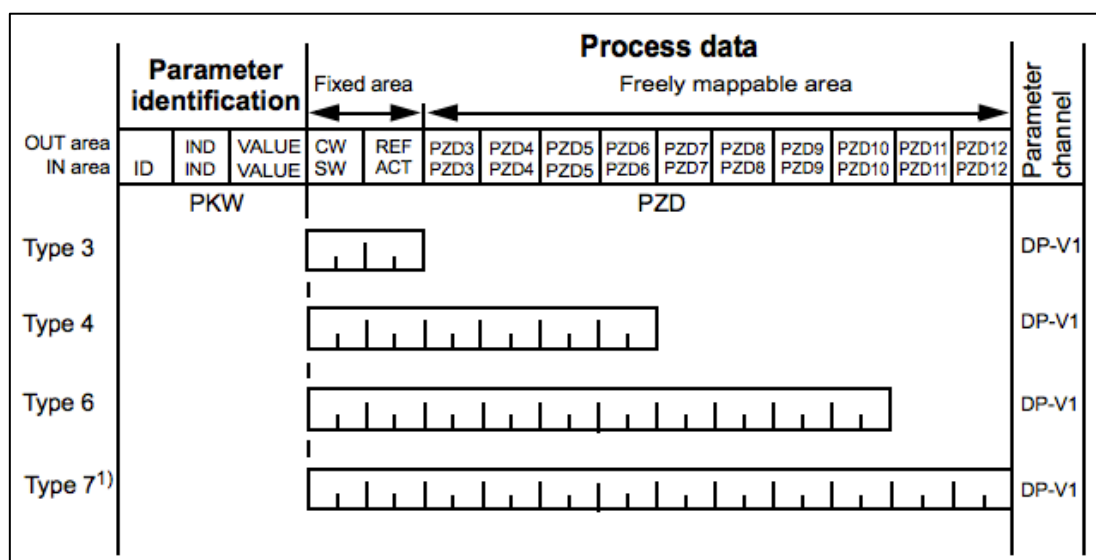
Kuva 9. Tiedonvaihto PLC:n ja taajuusmuuttajan välillä (ABB Industrial drives 2013)

Profinet-väylän kautta tapahtuvan ohjauksen standardoimiseksi Profinet-organisaatio on kehittänyt ns. ProfiDrive-profiileja. ProfiDrive-profiili määrittelee kenttäväyläohjauksessa olevien taajuusmuuttajien tiedonsiirrossa käytettävät PPO-tyypit (Parameter Process data Objects). PPO-tyyppejä hyödynnetään ohjelmoitavan logiikan ja taa-

juusmuuttajan välisessä tiedonsiirrossa. Kaikki PPO-tyypit on määritelty ohjelmoitavan logiikan ja taajuusmuuttajan välistä syklistä kommunikointia varten. (General Electric 2014.)

Käytännössä taajuusmuuttajan ja ohjelmoitavan logiikan välisessä syklisessä kommunikoinnissa laitteiden välillä liikkuu PPO-tyyppien määrittelemiä datasanomia. Syklistä kommunikointia käytettäessä dataliikenne koostuu 16- tai 32-bittisistä input- ja output-sanoista (Kuva 10). ACS880 Taajuusmuuttaja tukee enintään kahtatoista 16-bittistä datasanaa molempiin suuntiin väylällä. Datasanojen sisältö määritellään taajuusmuuttajan parametreissa. (ABB Industrial drives 2011.)

Jokainen IO-ohjaimen väylään lähettämä (OUT area) sanoma sisältää taajuusmuuttajan ohjaussanan (CW), pääohjearvon (REF) ja valitun PPO-tyypin määrittelemän määrän käyttäjän vapaasti käytettävissä olevia datasanoja (PZD). Esimerkiksi PPO Type 6 sisältää kahdeksan käyttäjän määriteltävissä olevaa datasanaa, joilla voidaan muuttaa esimerkiksi taajuusmuuttajan parametreja. Samoin jokainen IO-laitteen lähettämä vastaus (IN area) on saman PPO-tyypin mukainen datasanoma, joka sisältää taajuusmuuttajan tilasanan (SW), tilatiedon (ACT) sekä vastaavan määrän datasanoja (PZD), kuin taajuusmuuttajalle lähetetyssä sanomassa. (ABB Industrial drives 2011.)



Kuva 10. PPO-tyyppien sisältö (ABB Industrial drives 2011)

3.2.2 Ohjaus- ja tilasanat

Kenttäväyläohjauksessa taajuusmuuttaja liitetään väylään väyläadapterin avulla. Tällöin taajuusmuuttajaa kontrolloidaan ohjaus- ja tilasanojen sekä käyttäjän määrittelemien parametriseinojen avulla. Ohjauksena pitää sisällään kaikki taajuusmuuttajan käynnistämiseen ja pysäyttämiseen vaadittavat tiedot koodattuina 16-bittiseen saanaan. Tilasana puolestaan sisältää tietoa taajuusmuuttajan tilasta ja hetkellisistä oloarvoista. Ohjauksena muodostetaan Profinet-väylään liitettyssä IO-ohjaimessa eli ohjelmoitavassa logiikassa. Ohjauksena lähetetään IO-ohjaimelta taajuusmuuttajaan kenttäväyläadapterin kautta. Taajuusmuuttajassa muodostuva tilasana palauttaa sen hetkiset oloarvot ja tilan IO-ohjaimelle. (ABB Industrial drives 2011.)

3.2.3 Parametrit

Taajuusmuuttajan parametrit ovat käyttäjän määriteltävissä olevia arvoja, joita taajuusmuuttaja käyttää moottorin ohjaamisessa. Parametreilla määritellään myös laitteiston asetukset ja tiedot käytettävästä laitteista. Tässä projektissa keskeisimmät parametrit ovat käyttöliittymästä muutettavat moottoriparametrit ja Ethernet-verkkoadapterin asetukset.

4 AUTOMAATIOPROJEKTIN ELINKAARIVAIHEET

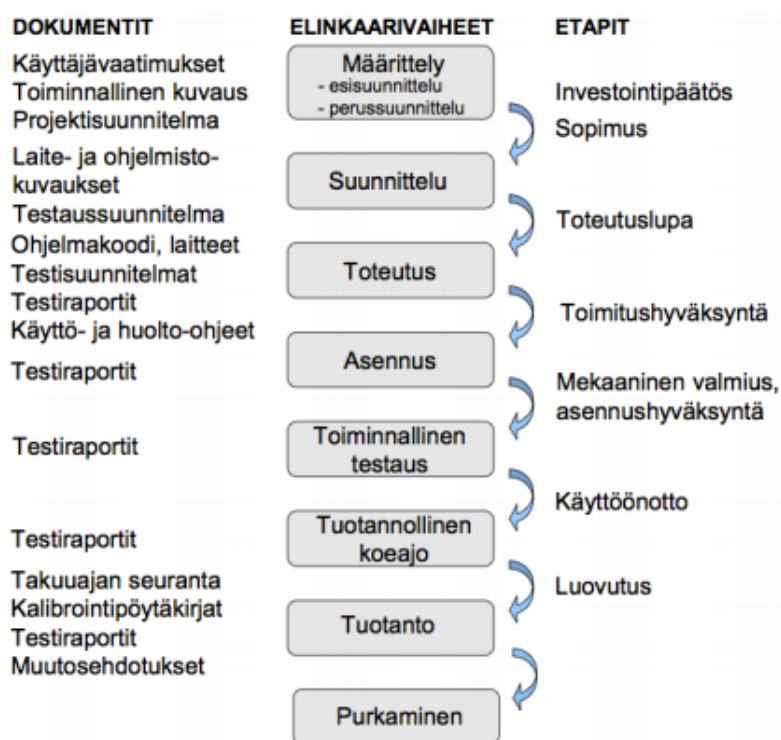
Automaatiojärjestelmän elinkaari voidaan karkeasti jakaa kolmeen eri vaiheeseen:

- Määrittelyvaihe
- Kehitysvaihe
- Tuotantovaihe

Määrittelyvaiheen tarkoitus on yksinkertaisesti selvittää mitä ollaan tekemässä. Tämän vaiheen päätehtävät ovat arkkitehtuurisuunnittelu, vaatimusten määrittely ja projektin suunnittelu. Kehitysvaiheeseen sen sijaan kuuluu järjestelmän toteuttaminen,

jonka päävaiheet ovat järjestelmän suunnittelu, toteutus ja testaus. Viimeisen vaiheen päätehtävä muutosten hallinta. Tässä vaiheessa korjataan asiakkaan löytämät virheet, mukautetaan järjestelmä uuteen ympäristöön sekä suunnitellaan ja toteutetaan järjestelmän jatkokehitys. Järjestelmän elinkaaren vaiheet on esitelty kuvassa 11, joka on sovitettu automaatioalalle. (Asmala ym. n.d., 31)

Kuvassa 11 esitetystä elinkaarimallista voidaan kuitenkin poiketa tarpeen mukaan. Esimerkiksi projektin laajuus, aikataulu ja yritysten omat toimintatavat saattavat muuttaa projektin kulkua tai laadittavia dokumentteja. Tässä automaatioprojektissa sovellettiin lähes kaikkia kuvan 11 elinkaarimallin vaiheita. Opinnäytetyöstä rajattiin pois muun muassa sähkökeskusten mekaaninen suunnittelu, sähkökeskusten koaminen ja käyttäjän löytämien virheiden korjaus. Tämän johdosta opinnäytetyöraportissa keskitytään tarkemmin järjestelmän suunnitteluun, logiikkaohjelman ja käyttöliittymän toteutukseen sekä testausvaiheeseen.



Kuva 11. Automaatioprojektin elinkaaren vaiheet (Asmala ym. n.d., 31)

5 AUTOMAATION OHJAUKSEN SUUNNITTELU

Automaation ohjauksen suunnitteluvaiheessa perehdyttiin rakennettavan järjestelmän tavoitteelliseen toimintaan ja rakenteeseen. Lisäksi käytiin keskusteluja asiakkaan kanssa laitteiston tarpeellisista ominaisuuksista ja vaatimuksista.

Järjestelmän tilauksesta laitetoimituksiin kulunut aika kului suurelta osin järjestelmään ja itselle vieraisiin ohjelmistoihin tutustumiseen. Ohjelmiin tutustumisen jälkeen käyttöliittymän ideointi ja logiikkaohjelman toteutus oli myös mahdollista aloittaa. Järjestelmän mekaaninen suunnittelu, sähkökeskusten kokoaminen sekä piirikaa-
vioiden suunnittelu laitevalintoineen toteutettiin Sataservice Oy:n puolesta.

5.1 Vaatimusten määrittely

Logiikkaohjelman vaatimukset pohdittiin toimeksiantajan kanssa. Käyttöliittymän käyttäjävaatimukset määriteltiin yhdessä asiakkaan kanssa alustavan suunnitelman mukaan.

Logiikkaohjelman vaatimukset:

- Jokainen taajuusmuuttaja käynnistetään omasta Start-painikkeesta ja kaikki pysäytetään yhdestä yhteisestä Stop-painikkeesta
- Käyttäjä valitsee käytettävän jännitteen ja käyttöpaikan fyysisillä valintakytkimillä
- Valintakytkinten tilatietojen perusteella asetetaan moottorilähdön nimellisen jännitteen suuruus taajuusmuuttajaan
- Taajuusmuuttajan käyttö estetään, mikäli halutun käyttöpaikan valintakytkimet on asetettu väärin
- Estetään liian pienten tai liian suurten moottoriparametrien käyttäminen käytöillä (Tehoalue oltava 20-100%)
- Integroidaan hätä-seis-toiminnot olemassa olevaan järjestelmään
- Estetään vahinkokäynnistyminen vikatilanteissa

Käyttöliittymän vaatimukset:

- Kosketusnäytöllä tulee olla käyttökohtaisesti havaittavissa vähintään moottorin ottaman virran ja tehon oloarvot
- Moottoriparametrit valitaan ja lähetetään taajuusmuuttajiin käyttökohtaisesti
- Parametrien lähetys estetään, jos taajuusmuuttaja on käytössä
- Vikatiedot näytetään käyttökohtaisesti
- Visuaalisesti helppokäyttöinen ja ymmärrettävä

5.2 Laitteistovalinnat

Kenttäväylä

Laitteiden ohjauksen ja parametrien tiedonsiirtotekniikaksi valittiin Profinet-kenttäväylä. Väylätekniikka todettiin parhaaksi ratkaisuksi, koska se on nopea ja erittäin monipuolinen. Kenttäväylän yksi suurimmista hyödyistä on mahdollisuus yhdistää monia laitteita saman väylän kautta. Vain jo tämä seikka voi vähentää jopa 40 % järjestelmän suunnittelusta, käyttöönotosta ja kunnossapidosta koituvista kustannuksista (PI North America 2011).

Suurimmat Profinet-kenttäväylän valintaan johtaneista syistä olivat järjestelmän vianhaun helppous ja mahdollisuus asettaa ja tarkastella parametreja yksinkertaisesti. Myös erilaisten lisäysten tekeminen jälkeenpäin on helpompaa. Lisäksi Profinet-väylä voidaan rakentaa lähes mihin tahansa Ethernet-verkkoon ilman erikoisliittimiä, toisin kuin esimerkiksi Profibus-väylä. (Tuomala sähköposti 19.3.2014.)

Ohjauspaneeli

Ohjauspaneeliksi valittiin Winmaten valmistama 10,4 ” kosketusnäytöllinen teollisuus-PC. Käyttöjärjestelmäksi tietokoneeseen valittiin sulautettu Windows 7 ja erillinen Indusoftin Web Studio valvomo-ohjelmisto. Tämä mahdollistaa valvomon korvaamisen paneelin vikaantuessa esimerkiksi kannettavalla tietokoneella. PC:n käyttö turvaa myös järjestelmän yhteensopivuuden tulevaisuudessa, koska se ei sido käyttämään tietyn valmistajan tuotteita. Ohjauspaneeli on koteloidu ruostumattomasta teräksestä tehtyyn koteloon, joten se soveltuu hyvin vaativiinkin teollisuusolosuhteisiin. (Tuomala sähköposti 19.3.2014.)

Yksi valintaan johtaneista seikoista oli myös PC:n edullisuus yksittäiseen 10 ” ohjauspaneeliin verrattuna. Kunnossapito on helpompaa, koska PC:n hajotessa tilalle asennetaan uusi PC, jolloin vain valvomo-ohjelmisto tulee päivittää ajantasalle. Windows 7 alustana mahdollistaa myös muiden sovellusten asentamisen, etätuen ja tarvittaessa pääsyn tehtaan muihin verkkoihin. (Tuomala sähköposti 19.3.2014.)

Taajuusmuuttajat

Taajuusmuuttajiksi valittiin ABB:n ACS880-sarjan laitteet. ACS880-sarjaan päädyttiin, koska Sataservicellä on jo vahva tuntemus ABB:n taajuusmuuttajista. Muita valintaan johtaneita seikkoja olivat taajuusmuuttajien ja niiden ohjauspaneelien helpokäyttöisyys, laitteiden hyvä tuki ja ominaisuudet. (Tuomala sähköposti 19.3.2014.)

Ohjelmoitava logiikka

Ohjelmoitavaksi logiikaksi valittiin ABB:n PM573-ETH. ABB:n logiikkaan päädyttiin, koska Sataservicellä on vahva tuntemus myös tämän valmistajan logiikoista. PM573-ETH -malli valittiin, koska projekti on kevyt, jolloin tarvitaan vähemmän muistia. Lisäksi logiikassa tuli olla sisäänrakennettu Ethernet-tuki. Yksi ratkaisevista tekijöistä oli myös laitteen edullisuus. (Tuomala sähköposti 19.3.2014.)

6 OHJELMOINTI

Logiikkaohjelma toteutettiin Control Builder 2.3 -ohjelmistolla. Control Builder on ABB Oy:n luoma työkalu, joka on kehitetty ABB:n logiikkaohjainten ja taajuusmuuttajien ohjelmointiin ja konfigurointiin IEC 61131-3 -standardiin kuuluvilla ohjelmointikielillä. Control Builder-ohjelmisto perustuu CodeSys-ohjelmointiympäristöön. (ABB OY:n www-sivut 2013).

6.1 Laitteistomäärittely

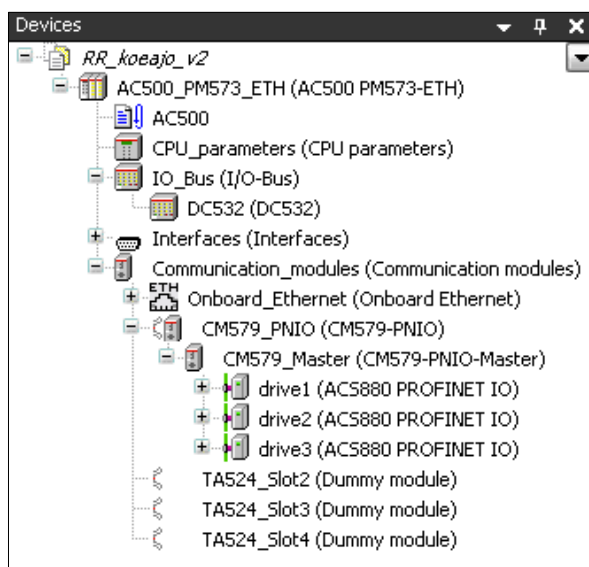
Uutta projektia aloitettaessa ensimmäisenä ohjelmalle tulee kertoa käytettävän PLC:n komponentit sekä tässä tapauksessa myös Profinet-väylä ja siihen liitettävät laitteet.

Control Builder -ohjelmistossa laitteiston kokoonpano määritellään aina ohjelman vasemmassa reunassa näkyvään Devices-ikkunaan (Kuva 12).

Tässä projektissa tarvittavat ohjelmoitavan logiikan komponentit ovat:

- Ohjelmoitava logiikka AC500 PM573_ETH
- Digitaalinen Input/Output-moduuli DC532
- Profinet-kommunikointimoduuli CM579
- Lisäksi määritellään väylään liitettävät laitteet

Kommunikointimoduuli CM579 mahdollistaa Profinet-väylän liittämisen osaksi automaatiojärjestelmää. Laitteet, joita halutaan ohjata väylän kautta, määritellään CM579_Master -komponentin alapuolelle. CM579_Master on luotavan väylän IO-ohjain. Drive1, Drive2 ja Drive3 ovat rakennettavan järjestelmän taajuusmuuttajat (IO-laitteet), joita kontrolloidaan Profinet-väylän avulla.



Kuva 12. Control Builder -ohjelmistoon määritelty laitteiston kokoonpano (Control Builder 2013)

6.2 Profinet-väylän käyttöönotto ja konfigurointi

Ennen varsinaisen ohjelman ja käyttöliittymän kehittämistä, kaikki väylän laitteet konfiguroitiin. Konfiguroinnin jälkeen väylän toiminta testattiin.

Ensimmäisenä tulee määritellä IP-osoitteet ja aliverkon peitteet niin, että PC, PLC ja taajuusmuuttajat ovat samassa verkossa. Vain samassa IP-osoiteavaruudessa olevat laitteet pystyvät kommunikoimaan keskenään. Tässä tapauksessa osoiteavaruutena käytettiin oletus osoiteavaruutta, joka alkaa osoitteesta 192.168.0.0 ja päättyy osoitteeseen 192.168.0.255. Aliverkon peitteeksi määriteltiin 255.255.255.0, koska tässä tapauksessa IP-verkkoa ei ollut tarvetta jakaa.

Seuraavaksi jokaisen taajuusmuuttajan Ethernet-adapteri otettiin käyttöön ja määriteltiin niiden asetukset. Asetukset määriteltiin käsin taajuusmuuttajien ohjauspaneelilla. Adapteri otetaan käyttöön valitsemalla taajuusmuuttajan asetuksista parametri-ryhmä, joka sisältää kenttäväyläsovittimen asetukset. Kenttäväyläsovitin eli Ethernet-adapteri otettiin käyttöön valitsemalla adapterin käytössä oleva fyysinen korttipaikka. Tämä mahdollistaa taajuusmuuttajan ohjaamisen kenttäväylän kautta. Lisäksi määriteltiin kenttäväyläadapterin tyyppi, sekä muut verkkoasetukset kuten IP-osoite, aliverkon peite, syklisen kommunikoinnin tyyppi ja tiedonsiirtohäiriön jälkeiset toimenpiteet. Lisäksi jokaiselle Profinet-verkossa olevalle IO-laitteelle annettiin yksilöllinen nimi Control Builder -ohjelmiston avulla.

Verkkoasetusten konfiguroinnin jälkeen määriteltiin taajuusmuuttajan kenttäväyläohjauksessa tarvittavat datasanomat. Näiden parametrin asettaminen mahdollistaa kommunikoinnin PLC:n ja taajuusmuuttajien välillä. Datatulo- ja lähtöparametrit määrittelevät syklisessä kommunikoinnissa päivitettävät datasanat. Datasanoja päivitetään jatkuvasti ja viive on erittäin lyhyt. PPO-tyypiksi valittiin PPO-tyyppi 6, koska se mahdollistaa kahdeksan eri valinnaisen datasanan siirtämisen ohjaus- tai tilasanan ja ohje- tai oloarvon lisäksi.

Datatulo parametriryhmään tulee asettaa ne sanat, jotka halutaan siirtää taajuusmuuttajasta kenttäväyläohjaimeen eli ohjelmoitavaan logiikkaan. Taajuusmuuttajaa ohjattaessa syklisen kommunikoinnin avulla, ensimmäinen sana varataan aina taajuusmuuttajan lähettämälle tilasalle ja seuraava sana ensimmäiselle oloarvolle (Kuva 10). Loput datasanat voidaan valita tarpeen mukaan.

Tässä tapauksessa nopeuden lisäksi taajuusmuuttajilta haluttiin seuraavat oloarvot näkyviin käyttöliittymään:

- Moottorin virta Ampeereina ja prosenttina nimellisvirrasta
- Moottorin ottama teho
- Moottorin nopeus
- Moottorin momentti
- Aktiivinen vikakoodi

Moottorin virran, virtaprocentin, tehon ja nopeuden oloarvot valittiin helpottamaan testauksen aikaista vianhakua. Näiden oloarvojen avulla on mahdollista havaita jo kosketusnäytöltä jos esimerkiksi testimoottori ottaa ylivirtaa, tai jumiutuneen moottorin momentti kasvaa liian suureksi. Aktiivinen vikakoodi vähentää myös käyttäjän työtä vikatilanteissa, koska käyttäjän ei tarvitse erikseen avata sähkökeskuksia selvittääkseen taajuusmuuttajan pysähtymisen syytä.

Datalähtö parametriryhmään puolestaan määritellään kenttäväyläohjaimesta taajuusmuuttajaan siirrettävät datasanat. PPO-tyypin mukaan myös tässä ensimmäinen sana varataan taajuusmuuttajan ohjaussanalle sekä toinen sana ohjearvolle (Kuva 10). Tämän laitteen tarkoitus on koekäyttää eri suuruisia moottoreita, jolloin moottoriparametrit tulee aina vaihtaa moottorikohtaisiksi.

Tämän johdosta seuraaviin siirrettäviin sanoihin valittiin moottorin konfigurointiparametrit:

- Nimellisjännite
- Nimellisvirta
- Nimellisnopeus
- Nimellisteho

Väylän toimivuus logiikalta taajuusmuuttajaan testattiin Control Builder –ohjelmistolla syöttämällä arvoja väylään lähetettävälle muuttujille (Kuva 13). Lisäksi taajuusmuuttajalta logiikkaan siirtyvä tieto testattiin valitsemalla taajuusmuuttajan parametreista datatuloon yksi datasana, joka piti sisällään esimerkiksi ympäristön lämpötilan. Näin varmistettiin tiedon kulku molempiin suuntiin väylässä. Väylän käyttöönotto sujui ongelmitta lukuun ottamatta yhtä rikkinäistä Ethernet-adapteria,

johon ei saatu aluksi yhteyttä lainkaan. Ongelmasta selvittiin vaihtamalla uusi Ethernet-adapteri laitteeseen.

Variable	Mapping	Channel	Address	Type	Unit	Description
drive1_SW		Input1	%IW1.0	UINT		Käyttö 1, Tilasana
drive1_Act1		Input2	%IW1.1	UINT		Käyttö 1, Taajuusmuuttajan nopeus
drive1_ActiveFault		Input3	%IW1.2	UINT		Käyttö1, Aktiivisen vian numero
drive1_ActPower		Input4	%IW1.3	UINT		Käyttö 1, Tehon seuranta käyttöliittymälle
drive1_ActCurrent		Input5	%IW1.4	UINT		Käyttö 1, Virranmittaus käyttöliittymälle
drive1_ActTorque		Input6	%IW1.5	UINT		Käyttö1, Momentin mittaus käyttöliittymälle prosentteina
		Input7	%IW1.6	UINT		
		Input8	%IW1.7	UINT		
		Input9	%IW1.8	UINT		
		Input10	%IW1.9	UINT		
drive1_CW		Output1	%QW1.0	UINT		Käyttö 1, Ohjaussana
drive1_Ref1		Output2	%QW1.1	UINT		Käyttö 1, Nopeusohje. Vaikka vakio nopeus, pakotetaan silti ohjearvo.
drive1_voltage_parameter		Output3	%QW1.2	UINT		Käyttö1, Nimellisjännite
drive1_current_parameter		Output4	%QW1.3	UINT		Käyttö1, Nimellisvirta
drive1_speed_parameter		Output5	%QW1.4	UINT		Käyttö1, Nimellisaika
drive1_power_parameter		Output6	%QW1.5	UINT		Käyttö1, Nimellisteho

Kuva 13. Väylään lähetettävät ja väylästä saapuvat datasanat tallennetaan erillisiin muuttujiin (Control Builder 2013)

6.3 Parametrien määrittäminen taajuusmuuttajaan

Profinet-verkkoadapterin käyttöönotossa määriteltyjen parametrien lisäksi taajuusmuuttajiin tuli asettaa muita projektikohtaisia parametreja. Parametrien säätäminen jatkui koko projektin ajan käyttöönottoon asti, koska alussa ei ollut varmuutta siitä, mitä kaikkia parametreja tulee asettaa. Muita määriteltäviä parametreja olivat esimerkiksi:

- Moottorin säätötapa
- Taajuusmuuttajan valvomat oloarvojen rajat
- Moottorin tyyppi, nimellistaajuus ja muut moottoriparametrit, joita ei muuteta käyttöliittymästä
- Ohje- ja oloarvojen asetukset
- Ohje- ja oloarvojen lähteet

6.4 Logiikkaohjelma

6.4.1 Ohjelmistoarkkitehtuuri

Ohjelmistoarkkitehtuuri kuvaa ohjelman rakennetta sekä eri osien välisiä suhteita. Arkkitehtuurit voidaan jakaa moniin erilaisiin ratkaisumalleihin. Tässä tapauksessa ohjelmistoarkkitehtuuriksi valittiin kutsumekanismiarkkitehtuuri. Kutsumekanismiarkkitehtuuri koostuu pääohjelmasta ja aliohjelmista. Kutsumekanismiarkkitehtuurissa jokaista aliohjelmaa ohjataan ylemmältä tasolta. (Asmala 2013.)

Kutsumekanismiarkkitehtuuriin päädyttiin, koska järjestelmässä tarvitaan useampaan kertaan samoja aliohjelmia. Samoja yleiskäyttöisiä funktioita voidaan hyödyntää kolmen identtisen taajuusmuuttajan ohjauksissa ja parametrikyselyissä. Lisäksi haluttiin helpottaa tulevaisuudessa tehtävää kehitystyötä ja vianhakua. Esimerkiksi yhteen yleiskäyttöiseen funktioon tehty muutos päivittyy automaattisesti kaikkiin ohjelma-kohtiin, joissa kyseistä funktiota käytetään.

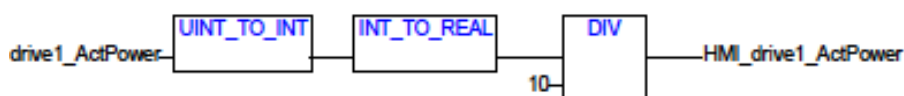
6.4.2 Pääohjelma

Pääohjelmaan eli logiikassa syklisesti suoritettavaan ohjelmaan ohjelmoitiin jokaisen käytön ohjaus ja oloarvojen seuranta (Liite 1). Tämän lisäksi sykliseen ohjelmaan ohjelmoitiin taajuusmuuttajien, hätä-seis-piirin ja Profinet-verkon vikatietojen siirtäminen käyttöliittymään.

Taajuusmuuttajien vikatieto saatiin suoraan väylästä, koska taajuusmuuttajasta saapuva tilasana pitää sisällään vikabitin. Hätä-seis-piirin vikatieto puolestaan saatiin toisesta hätä-seis-järjestelmästä. Profinet-verkon tilan tarkkailu toteutettiin Profinet-kirjastosta löytyvällä PNIO_STATE-lohkolla (Liite 1). Tämän lohkon COMM_ERNO-tieto kertoo sen hetkisen Profinet-verkon tilan. Jos COMM_ERNO=0, verkko on normaalissa toimintatilassa. Tätä tietoa hyödynnetään sähkökatkon jälkeen, kun parametrit lähetetään taajuusmuuttajiin. Logiikan patterin muistissa olevat edelliset käyttäjän antamat parametrit lähetetään sähkökatkosta palautumisen jälkeen

automaattisesti, kun Profinet-verkko on jälleen kunnossa. COMM_ERNO-tietoa käytetään lisäksi Profinet-verkon häiriökuvakkeen näyttämiseen käyttöliittymässä.

Väylässä liikkuva tieto on UINT-tyyppistä, joten tämä tuli huomioida tietoa lähetettäessä tai vastaanotettaessa väylästä. UINT-tietotyyppi on 16-bittinen etumerkitön Integer-datatyyppi. UINT-tietotyypin arvo voi vaihdella välillä 0 – 65535. Taajuusmuuttajilta saatavat oloarvot tuli myös skaalata oikeaan suhteeseen. Esimerkiksi väylästä saatava hetkellinen teho ”drive1_ActPower” muunnettiin ensin UINT-tietotyyppistä INT-tietotyyppin, jonka jälkeen REAL-tietotyyppiksi. Tämän jälkeen luku jaettiin kymmenellä, jolloin se vastasi todellista moottorin ottamaa tehoa. Lopullinen arvo siirrettiin erilliseen muuttujaan, jonka arvoa näytetään käyttäjälle (Kuva 14).



Kuva 14. Hetkellisen tehon oloarvon muunnos ja skaalaus käyttöliittymässä monitorointia varten (Control Builder 2013)

Oloarvojen lisäksi käytöille pakotettiin ohjearvo ”Ref1”. Koska sähkömoottoreita koestetaan täydellä nopeudella, taajuusohje pakotettiin 100 prosenttiin. Taajuusohjeen skaalaus toteutettiin kuitenkin niin, että sitä on mahdollista muuttaa helposti (Liite 5). Taajuusohjeen prosentti, -100 – 0 – 100 skaalattiin vastaamaan lukuja -20000 – 0 – 20000. Tämän skaalauksen avulla oli mahdollista kääntää moottorin pyörimissuunta vaihtamalla ohjearvon etumerkkiä. Tarvittaessa myös taajuusohjetta voidaan muokata.

Muut tarvittavat skaalaukset tehtiin pääohjelmaan (Liite 1) tai erillisiin funktioihin, kuten nopeuden skaalaus (Liite 6) ja virtaprocentin laskenta (Liite 7).

6.4.3 Ohjaus- ja tilasanojen käsittely

Taajuusmuuttajan ohjauksessa tarvittavien ohjaus- ja tilasanojen käsittely ohjelmoitiin erillisiin funktioihin (Liite 2). Näiden funktioiden tarkoitus on purkaa väylästä

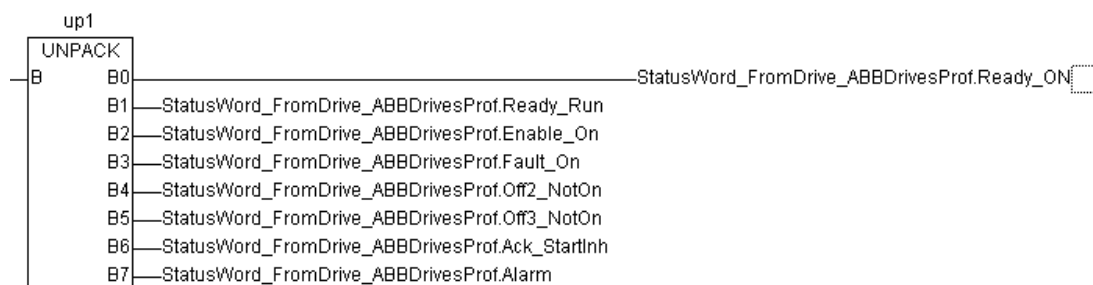
saapuva tilasana biteiksi ja pakata lähtevä ohjaussana biteistä WORD-tietotyyppiin. Näiden funktioiden avulla taajuusmuuttajan ohjelmointia voidaan helpottaa huomattavasti eikä ohjelmoijan tarvitse muistaa tiettyjä bittijärjestyksiä, numerojonoja tai heksalukuja. Esimerkiksi taajuusmuuttajan tilasana sisältää 16 bittiä, joista bitit 10 - 14 ovat käyttäjän määriteltävissä. Tämän johdosta käyttäjän lähes mahdoton muistaa ulkoa eri bittien merkitystä (Kuva 15).

Bit	Name	Value	STATE/Description
0	Ready to switch ON	1	READY TO SWITCH ON.
		0	NOT READY TO SWITCH ON.
1	Ready run	1	READY TO OPERATE.
		0	OFF1 ACTIVE.
2	Ready ref	1	OPERATION ENABLED.
		0	OPERATION INHIBITED.
3	Tripped	1	FAULT.
		0	No fault.
4	Off 2 inactive	1	OFF2 inactive.
		0	OFF2 ACTIVE.
5	Off 3 inactive	1	OFF3 inactive.
		0	OFF3 ACTIVE.
6	Switch-on inhibited	1	SWITCH-ON INHIBITED.
		0	–
7	Warning	1	Warning active.
		0	No warning active.
8	At setpoint	1	OPERATING. Actual value equals reference = is within tolerance limits (see parameters 46.21...46.23).
		0	Actual value differs from reference = is outside tolerance limits.
9	Remote	1	Drive control location: REMOTE (EXT1 or EXT2).
		0	Drive control location: LOCAL.
10	Above limit	-	See parameter 06.29 MSW bit 10 sel.
11	User bit 0	-	See parameter 06.30 MSW bit 11 sel.
12	User bit 1	-	See parameter 06.31 MSW bit 12 sel.
13	User bit 2	-	See parameter 06.32 MSW bit 13 sel.
14	User bit 3	-	See parameter 06.33 MSW bit 14 sel.
15	Reserved		

Kuva 15. Taajuusmuuttajalta ohjelmoitavaan logiikkaan siirrettävä tilasana koostuu 16:sta eri bitistä.

Esimerkiksi väylästä saapuva tilasana puretaan ensin biteiksi, jonka jälkeen siirretään käyttäjän määrittelemään globaaliin ”Status-Word_FromDrive_ABBDrivesProf”-tietotyyppiin. Tämä tietotyyppi on itse määritelty STRUCT-rakenne, johon on määritelty vain tilasanan sisältämät bitit. Tämän kaltaisen itse tehdyn rakenteen käyttäminen mahdollistaa tilasanan tiettyjen bittien monitoroinnin suoraan viittaamalla STRUCT-tietotyyppiin. Lisäksi ohjaus- ja tilasanojen bittejä on mahdollista käyttää suoraan taajuusmuuttajan ohjaamisessa. Esimerkiksi taajuusmuuttajan vikatilaa on mahdollista seurata monitoroimalla suoraan globaali-

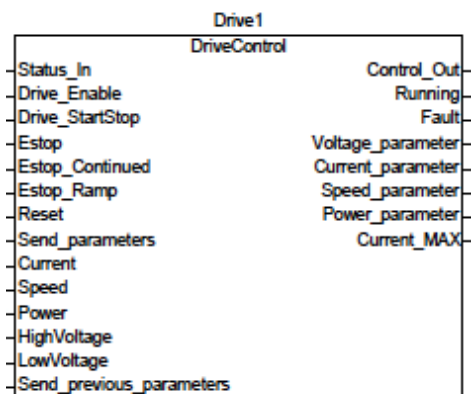
lin ”StatusWord_FromDrive_ABBDrivesProf”-rakenteen muuttujaa ”Fault_On” (Kuva 16).



Kuva 16. Tilasan ensimmäiset kahdeksan bittia puretaan up1-lohkossa ja siirretään omaan tietotyyppiin (Control Builder 2013)

6.4.4 Taajuusmuuttajan ohjaus

Taajuusmuuttajan ohjaus rakennettiin yhteen erilliseen ”DriveControl”-funktioon (Kuva 17). Yhden yleiskäyttöisen funktion käyttäminen mahdollistaa saman lohkon käyttämisen muidenkin taajuusmuuttajien ohjaamisessa (Liite 3).



Kuva 17. Taajuusmuuttajan ohjausfunktion kutsu (Control Builder 2013)

Ohjausfunktioon toteutettiin taajuusmuuttajan käynnistys- ja pysäytystoiminto, vian kuittaus, hätä-seis-toiminto sekä moottoriparametrien lähetys taajuusmuuttajaan. Väylältä saatavan laitteen tilasanan bittejä hyödyntämällä voidaan valvoa myös laitteen käynti- ja häiriötietoja.

Taajuusmuuttajan ajoitiksi ohjelmoitiin ”Drive_StartStop”-bitti. Käyttäjän tekemien väärin jännitteenvalintakytkinten asentojen ja mahdollisten kytkinten vikatilantei-

den johdosta taajuusmuuttajan käynnistämisehdoksi kehitettiin erillinen lukitusbitti ”Drive_Enable”. ”Drive_Enable”-bitin tulee olla päällä, ennen kuin taajuusmuuttajaa on mahdollista ajaa. Lukitusbitin tarkoitus on estää moottorin käynnistäminen, jos jompikumpi edellä mainituista tilanteista toteutuu. Toisin sanoen, käyttäjän tulee valita ensin jännitteenvalintakytkimet oikein, ennen kuin taajuusmuuttajaa on mahdollista käyttää.

Taajuusmuuttajan ohjausfunktioon toteutettiin myös moottoriparametrien siirtäminen väylän kautta taajuusmuuttajiin. Käyttöliittymästä annetut moottoriparametrit tuodaan ohjausfunktion tuloihin ”Current”, ”Speed” ja ”Power” (Kuva 17). Turvallisuussyistä, käytettävä jännite valitaan fyysisesti jännitteenvalintakytkimillä. Näiden kytkinten tilatiedoilla valitaan käytettävä jännite, joten käyttäjälle ei anneta edes mahdollisuutta valita jänniteparametria käyttöliittymästä. Tämä estää esimerkiksi 690 VAC jännitteen syöttämisen 400 VAC sähkömoottoriin. Esimerkiksi moottorin rikkoutuminen voidaan välttää tilanteessa, jossa käyttäjä valitsee kytkimillä tahattomasti 400 VAC jännitteen ja käyttöpaikan, jonka jälkeen käyttäjä antaa parametreiksi 690 VAC moottorin tiedot. Kytkinten tilatiedot tuodaan ohjausfunktion tuloihin ”HighVoltage” ja ”LowVoltage” (Kuva 17). Käyttäjän painaessa ”Lähetä parametrit”-painiketta käyttöliittymästä, nämä parametrit korvaavat taajuusmuuttajassa olevat moottoriparametrit. Näiden neljän moottoriparametrin lisäksi laitteeseen lähetetään valittua virtaparametria vastaava maksimivirta ”CurrentMAX”. Taajuusmuuttajan sallima moottorin maksimivirta lasketaan kertomalla käyttäjän antaman virtaparametrin arvo 1.6:lla. Turvallisuussyistä myös parametrien lähetys on estetty, jos taajuusmuuttaja on käytössä.

Parametrit voidaan lähettää myös ”Send_previous_parameters”-bitin avulla. Tämän bitin tarkoitus on lähettää logiikan patterin avulla muistissa olevat käyttäjän antamat edelliset parametrit sähkökatkon jälkeen.

6.4.5 Parametrien käsittely

PPO-tyyppi 6 soveltui hyvin taajuusmuuttajan ohjaamisen lisäksi siirtämään tarvittavat moottoriparametrien muutokset taajuusmuuttajiin. Sen sijaan taajuusmuuttajalta

takaisin logiikkaan siirrettävät datasanat loppuivat kesken, koska valittujen oloarvojen lisäksi käyttöliittymään haluttiin tietoa onnistuneesta parametrien vaihdosta. Käytännössä taajuusmuuttajilta haluttiin lukea vaihdettujen moottoriparametrien arvot, joka olisi vaatinut lisää vapaita datasanoja. Tämän johdosta parametrien lukeminen toteutettiin asyklisella parametrikyselyllä.

Asyklista parametrikyselyä varten tehtiin erillinen funktio (Liite 4), joka lukee valitut parametrit halutusta taajuusmuuttajasta. Tätä aliohjelmaa kutsutaan pääohjelmassa aina, kun käyttäjä lähettää parametrit käyttöliittymästä. Käyttäjän painaessa ”Lähetä parametrit”-painiketta, funktio aloittaa kyselyprosessin. Ensimmäisenä funktio kirjoittaa väylään kyselyn, voiko parametrikyselyn aloittaa. Onnistuneen kyselyn jälkeen lähetetään varsinainen parametrien lukukysely väylään. Lukukysely palauttaa parametrien tiedot WORD-tietotyypeistä koostuvaa taulukkoon. Lopuksi taulukkoa hyödyntämällä siirretään parametrien arvot käyttöliittymään, jolloin käyttäjä näkee parametrien vaihdon onnistuneen. Tämä parametrien kyselyprosessi toteutetaan myös sähkökatkosta palautumisen jälkeen, jotta käyttäjä näkee ajantasaisen tiedon laitteissa olevista parametreista.

Parametrien kyselyprosessin vikatilanteita varten kehitettiin erillinen häiriöbitti. Häiriöbitti asetetaan päälle, jos kirjoitus- tai lukukyselyissä ilmenee virhe, tai prosessin ”Tila”-muuttuja jää jumiin. Ilmenneestä häiriöstä näytetään häiriökuvake käyttöliittymässä.

6.5 Käyttöliittymän kehittäminen

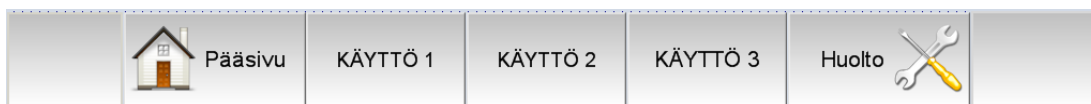
Käyttöliittymän kehittämisen perusajatuksena oli tehdä käyttöliittymästä mahdollisimman yksinkertainen ja selkeä. Käyttöliittymän tekeminen alkoi ennestään vieraan ohjelman opettelulla.

6.5.1 Käyttöliittymän rakenne

Ensimmäisenä oli luonnollista hahmotella koko käyttöliittymän looginen toiminta. Käyttöliittymään suunniteltiin yksi pääsivu sekä jokaiselle käytölle oma sivu. Pääsi-

vun tarkoitus on toimia etusivuna, kun järjestelmä avataan. Lisäksi pääsivulla näytetään jokaisen käytön hetkelliset oloarvot ja kaikki häiriötiedot. Käyttökohtaisten sivujen päätarkoitus on parametrien vaihtaminen. Tämän lisäksi käyttökohtaisilla sivuilla on nähtävissä kyseisen taajuusmuuttajan häiriötieto ja aktiivisen vian numero.

Käyttöliittymän tulee olla käytettävyydeltään mahdollisimman helppokäyttöinen, ymmärrettävä ja tarkoituksenmukainen (Heimbürger ym. 2011, 103). Tämän johdosta käyttöliittymään tehtiin jokaisella sivulla näkyvä valikkorivi (Kuva 18). Valikkorivin avulla on mahdollista siirtyä sivulta toiselle vaivattomasti. Pääsivun ja huoltosivun tunnistamista varten painikkeisiin asetettiin symbolit kuvamaan tiettyjä toimintoja. Kaikki symbolit ja kuvat ovat joko kaikkien käytettävissä olevan vapaan lisenssin alaisia tai itse tehtyjä. Kaikki symbolit ovat myös vektorigrafiikkaa, joten jälkepäin niiden muokkaaminen ja esimerkiksi suurentaminen on mahdollista ilman kuvakkeen pikselöitymistä.



Kuva 18. Jatkuvasti näkyvä valikkorivi näytön alareunassa (Web Studio 2013)

Sivulta toiselle siirtyminen erillisen valikkorivin kanssa toteutettiin jakamalla käytettävissä oleva tila kahteen osaan. Käytettävissä oleva tila määriteltiin projektissa käytettävän kosketusnäytön resoluution (1024x768) mukaan. Tila jaettiin kahteen osioon luomalla valikkoriville yksi osio (1024x100 pikseliä) ja varsinaisille sivuille toinen osio (1024x668). Käynnistyksessä aukeamaan asetettiin pääsivu sekä valikkorivi, jolloin näytöllä on näkyvissä yksi yhtenäinen ja siisti sivu (Liite 8). Tämän lisäksi luotiin oma sivu jokaiselle käytölle ja huoltotoiminnalle. Sivulta toiselle siirtyminen aktivoi aina halutun sivun näkyviin valikkorivin yläpuolella olevaan tilaan, jolloin valikko on aina käytettävissä.

6.5.2 Pääsivu

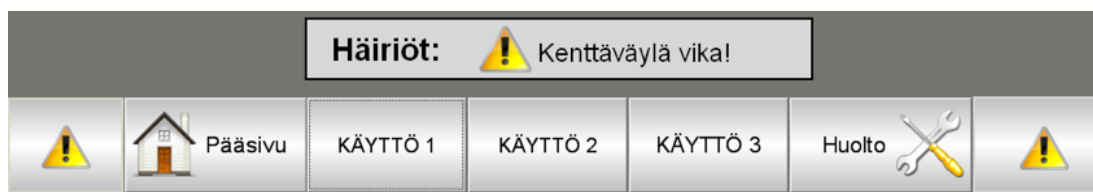
Pääsivun tehtävä on toimia etusivuna järjestelmälle ja näyttää jokaisen käytön oloarvot ja taajuusmuuttajan käyntitieto käyttäjälle (Liite 8). Etusivu jaettiin kolmeen eri

osioon, joista jokainen otsikoitiin käyttökohtaisesti. Osioihin jakaminen ja nimeäminen selkeyttää käyttöliittymän käyttämistä ja ymmärrettävyyttä.

Vaatimusten määrittelyssä aiemmin valitut oloarvot ja taajuusmuuttajan ”KÄY”-tieto sijoitettiin keskeiselle paikalle käyttäjän nähtäväksi. Lisäksi käyttökohtaisiin osioihin lisättiin häiriötiedon näyttäminen käyttäjälle. Taajuusmuuttajan vikaantuessa, käyttäjälle vilkutetaan sekunnin välein häiriökuvaketta käyttökohtaisen osion yläreunassa. Näin käyttäjän huomio saadaan heti kiinnitettyä oikeaan taajuusmuuttajaan ja ohjataan avaamaan vikaantuneen taajuusmuuttajan käyttökohtainen sivu.

Oloarvojen lisäksi käyttökohtaisiin osioihin toteutettiin käyttäjän valitseman jännitteen ja käyttöpaikan informaatio. Tiedot perustuvat jännitteenvaihtokytkinten tilatietoihin ja näin käyttäjä voi varmistaa kytkinten oikeellisuuden jo käyttöliittymästä käsin.

Pääsivun alareunaan suunniteltiin erillinen ”Häiriöt”-osio ja alavalikkoon lisättiin pienemmät häiriökuvakkeet (Kuva 19). Näiden tarkoitus on kiinnittää käyttäjän huomio, näyttää häiriökuvaketta ja ilmaista häiriön kohde. Käyttöliittymän alavalikkoon toteutetut häiriökuvakkeet vilkkuvat aina, kun jokin häiriö on päällä koeajojärjestelmässä. Näin käyttäjän huomio saadaan kiinnitettyä myös esimerkiksi siinä tilanteessa, jossa huoltosivu olisi tahattomasti jäänyt auki.



Kuva 19. Kenttäväylässä ilmenneen häiriön näyttäminen käyttöliittymässä (Web Studio 2013)

6.5.3 Käyttökohtaiset sivut

Käyttökohtaisten sivujen päätarkoitus on toimia käyttäjälle ”oikopolkuna” taajuusmuuttajan asetuksiin. Pääsivun tavoin myös käyttökohtaiset sivut on jaettu eri osaluokkiin hahmottamisen helpottamiseksi. Parametrien muuttaminen ja hälytystiedot

ovat erotettu toisistaan. Lisäksi parametrien lähettäminen ja taajuusmuuttajassa käytössä olevat parametrit on erotettu toisistaan. Lisäksi sivulle lisättiin nuoli kuvaamaan parametrien siirtymistä laitteeseen. Nämä seikat selkeyttävät käyttökohtaista sivua ja pitävät sivun luettavuuden hyvänä (Liite 8).

Parametreja muutettaessa käyttäjän antamat parametrien arvot kirjoitetaan logiikan muuttujiin. Lopuksi käyttäjä painaa ”Lähetä parametrit”-painiketta, jolloin muuttujiin tallentuneet parametrien arvot lähetetään taajuusmuuttajaan. Käyttäjälle tulee aina antaa jokin tieto siitä, että toimenpide on onnistunut (Heimbürger ym. 2011, 112). ”Käyttäjän ei pidä joutua kysymään: ’Menikö se perille?’, ’Tekeekö se nyt jotain?’ tai ’Tuliko tämä nyt tehdyksi?’ ” (Heimbürger ym. 2011, 112). Tällä sivulla käyttäjä havaitsee välittömästi jo painikkeen ”painautumisesta”, että komento otetaan vastaan. Tämän lisäksi ”Lähetä parametrit”-painikkeen painallus aloittaa logiikassa parametrikyselyn taajuusmuuttajalle ja tulostaa käyttöliittymään käytössä olevien parametrien arvot. Näin käyttäjän on heti mahdollista varmistaa, että halutut parametrit siirtyivät laitteeseen. Parametrikyselyn tulokset saadaan alle sekunnin viiveellä, joten käyttäjä ehtii yhdistämään painikkeen painamisen ja vieressä näkyvät tulokset toisiinsa. Tämän johdosta erillistä ilmoitusta onnistumisesta ei lisätty. Parametrikyselyn epäonnistumisesta sen sijaan toteutettiin erillinen häiriöilmoitus. Häiriöilmoitus näytetään, jos parametrikyselyprosessissa ilmenee vika tai tuloksia ei saada viiden sekunnin sisällä. Häiriöilmoituksen ohessa opastetaan käyttäjää lähettämään parametrit uudelleen.

Parametrien vaihtamisen lisäksi käyttökohtainen sivu antaa käyttäjälle tietoa kyseisestä käytöstä. Sivun alareunaan luodulla erillisellä alueella näytetään vilkkuva varoituskolmio, taajuusmuuttajan mennessä vikatilaan. Samalla käyttäjälle näytetään vikakoodin numero ja opastetaan avaamaan linkki, josta löytyvät kaikki vikakoodien selitykset (Kuva 20). Vikakoodien selitykset koottiin laitemanuaalista erilliseen PDF-tiedostoon. Info-kuvaketta painettaessa käyttöliittymä avaa hyperlinkin, joka on osoitettu PDF-tiedostoon. PDF-tiedosto aukeaa kosketusnäytön Windows-käyttöjärjestelmään asennetussa PDF-selaimessa.



Kuva 20. Taajuusmuuttajan viankuittaus ja vikakoodi käyttökohtaisella sivulla (Web Studio 2013)

Näiden toimintojen lisäksi käyttökohtaisille sivuille päätettiin toteuttaa taajuusmuuttajien toiminta-arvojen rajojen valvonta. Järjestelmän vaatimuksissa määriteltiin, että käyttäjän estetään valitsemasta liian pieniä tai liian suuria moottoriparametreja. Tehoalueen tuli olla 20-100 % taajuusmuuttajan nimellisarvoista. Raja-arvojen seuranta toteutettiin Web Studio -ohjelmasta löytyvillä ominaisuuksilla. Käyttäjän painaessa muokattavaa parametria, aukeaa erillinen numeronäppäimistö, jonka yhteydessä näytetään kyseisen parametrin MIN ja MAX arvot. Käyttöliittymä ei anna käyttäjän valita näitä arvoja suurempia lukuja, ja näin on mahdollista valvoa käytöille annettavia moottoriparametrien arvoja. Esimerkiksi käyttäjän antaessa liian suuren moottoriparametrin arvon, järjestelmä antaa huomautuksen ja kertoo mikä on suurin mahdollinen arvo. Käyttäjän painaessa ”OK”-painiketta, järjestelmä pyytää uutta parametrin arvoa.

6.5.4 Huoltosivu

Huoltosivun päätarkoitus on valvomo-ohjelmiston sulkeminen. Valvomon sulkeminen on mahdollista vain ”Sulje valvomo”-painikkeella. Painike on sijoitettu erilliselle huoltosivulle, koska tarkoitus minimoida käyttäjän mahdollisuus sulkea valvomo vahingossa.

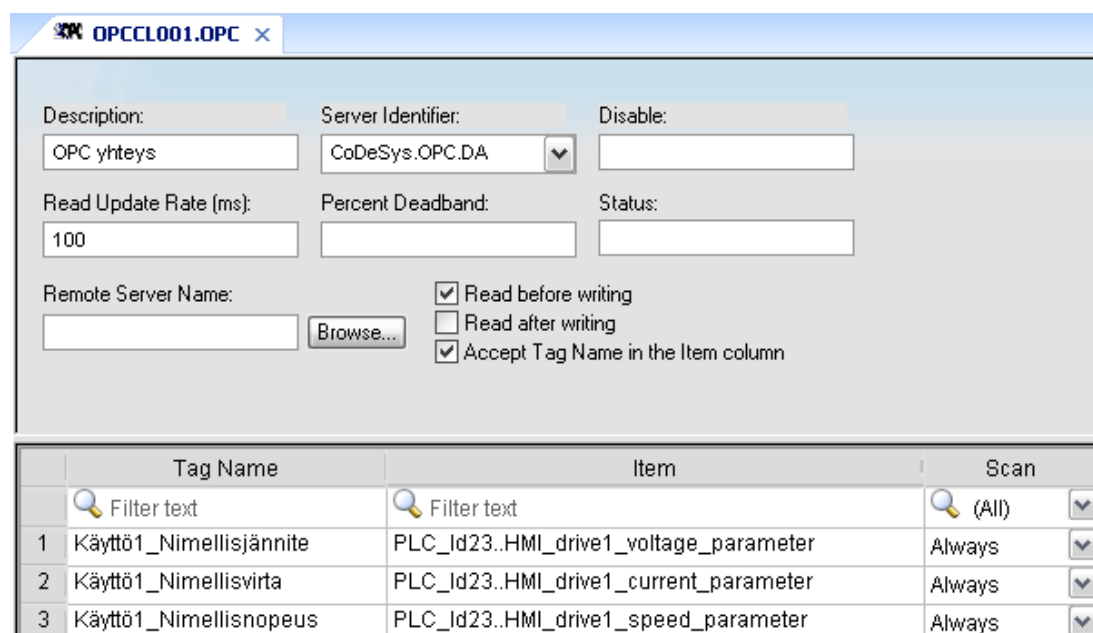
6.5.5 Käyttöliittymän integrointi logiikkaan

Tagit ovat yksi käyttöliittymän kehittämisen ydinkomponenteista. Tagit ovat muuttujia, joita käytetään valvomo-ohjelmistoissa käyttöliittymien suunnitteluun. Web Studio käyttää tageja vastaanottaakseen ja tallentaakseen tietoa logiikalta, erilaisista laskutoimituksista, funktioista sekä käyttäjältä. Toisaalta, tageja käytetään tiedon näyt-

tämiseen näytöillä, Internet-sivuilla, muokkaamaan näytön eri objekteja sekä kontrolloimaan eri käytönaikaisia tehtäviä. (Web Studio 2013.)

Yhdistämällä logiikan muuttujalistat Web Studion kanssa, on mahdollista monitoroida ja kontrolloida logiikan muuttujia käyttöliittymän tagien avulla. Web Studiassa käyttöliittymä integroidaan käytössä olevaan logiikkaan valitsemalla tiedonsiirtotapa ja asentamalla valitun tiedonsiirtotavan ajurit käytettäviin laitteisiin. (Web Studio 2013.)

Tiedonsiirtotavaksi valittiin OPC sen helppokäyttöisyyden vuoksi. OPC on avoimeen lähdekoodiin perustuva tiedonsiirtomalli, jota käytetään useimmiten ohjelmoitavien logiikoiden ja valvomo-ohjelmiston väliseen tiedonsiirtoon. Tiedonsiirto OPC:n kautta aloitetaan asentamalla ajurit käyttöliittymää ajavaan sulautettuun käyttöjärjestelmään ja ohjelmoinnissa käytettävään tietokoneeseen. Ajureiden asentamisen yhteydessä valitaan Control Builder -ohjelmistosta jaettavat muuttujakirjastot. Muuttujakirjastoista valitaan ne, jotka halutaan näkyvän valvomo-ohjelmistossa. Asentamisen jälkeen muuttujat näkyvät Web Studion OPC-valikossa (Kuva 21).



Kuva 21. Indusoft Web Studio-ohjelmisto, OPC-yhteyden välilehti ja muuttujalista (Web Studio 2013)

7 KÄYTTÖÖNOTTO

Koekäyttöjärjestelmä testattiin mahdollisimman pitkälle ensin toimeksiantajan tiloissa, jonka jälkeen järjestelmä asennettiin asiakkaan tiloihin. Viimeiset testaukset ja käyttöönotto suoritettiin asiakkaan tiloissa.

7.1 Järjestelmän testaus

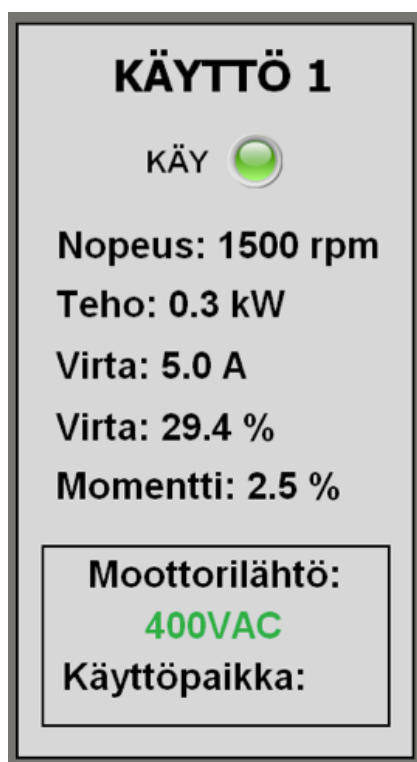
Laitteiston testaus aloitettiin jo ennen kuin sähkökeskuksia oli koottu valmiiksi. Keskukseseen kytkettiin väliaikainen 400 VAC ohjausjännite. Tässä vaiheessa oli mahdollista koestaa väylän toiminta syklisen ja asyklisen tiedonsiirrontekniikan avulla. Väylän syklinen toiminta testattiin aikaisemmin tässä dokumentissa kuvatulla tavalla. Asyklinen kommunikointia oli mahdollista testata samalla, kun asyklisen parametrikyselyn funktiota ohjelmoitiin. Parametrikyselyn toiminta oli melko haastava ymmärtää, mutta ABB-yritykseltä saatujen ohjelmointivinkkien avulla ja manuaalia tutkimalla parametrikysely kuitenkin saatiin toimimaan. Parametrikyselyfunktion toimintaa testattiin lähettämällä parametreja käyttöliittymästä ja näiden toimintojen testaus jatkui projektin loppuun saakka. Ohjelmoitavan logiikan IO-kortti määriteltiin ohjelmaan vasta tässä vaiheessa, koska kaikki järjestelmään tulevat tulot ja lähdöt eivät olleet aikaisemmin selvillä.

Sähkökeskukset koottiin valmiiksi, jonka jälkeen järjestelmään kytkettiin väliaikaisesti 690 VAC pääjännite ja 400 VAC ohjausjännite testausta varten. Lisäksi järjestelmään kytkettiin yksi 11 kW testimoottori. Kiireellisen aikataulun vuoksi, väylän ja parametrikyselyjen testaus sekä logiikkaohjelman ohjelmointi päätettiin tehdä mahdollisimman valmiiksi vain yhteen taajuusmuuttajakäyttöön, koska muut käytöt olivat ohjaukseltaan identtisiä.

Ensimmäisenä varmistettiin väylän toiminta. Parametrien siirtoa ja parametrikyselyn toiminnan testausta jatkettiin lähettämällä eri arvoja kosketusnäytöltä taajuusmuuttajaan. Tässä vaiheessa ei enää ilmennyt merkittäviä ongelmia, koska väylän toimintaa oli mahdollista testata jo aikaisemmin. Seuraavaksi testattiin taajuusmuuttajan ohjaus. Ennen koestamista, ohjaukseen lisättiin taajuusohjeen skaalaus. Testivaiheessa moottoria testattiin ensin pienellä nopeudella ja myöhemmin antamalla eri moottori-

parametreja käyttöliittymästä. Testaus jouduttiin tekemään vain yhdellä jännitteellä, ja lähes samoilla parametrien arvoilla, koska erilaisia testimoottoreita ei ollut saatavilla. Varsinainen moottorinohjaus erillisen käsiohjaimen avulla toimi kuitenkin ilman erityisiä ongelmia, vaikka testimoottoreita oli vain yksi.

Moottorin toimiessa oli mahdollista viimeistellä väylästä saatavien oloarvojen skaalaukset. Oloarvojen oikeellisuus tarkasteltiin ajamalla moottoria, jolloin käyttöliittymässä ja taajuusmuuttajan ohjauspaneelissa näkyviä oloarvoja oli mahdollista vertailla (Kuva 22). Tämän vertailun pohjalta logiikkaohjelmaan tehtiin oloarvojen skaalausmuutoksia.



Kuva 22. Käyttöliittymän hetkellisten oloarvot testattiin ajamalla moottoria (Web Studio 2013)

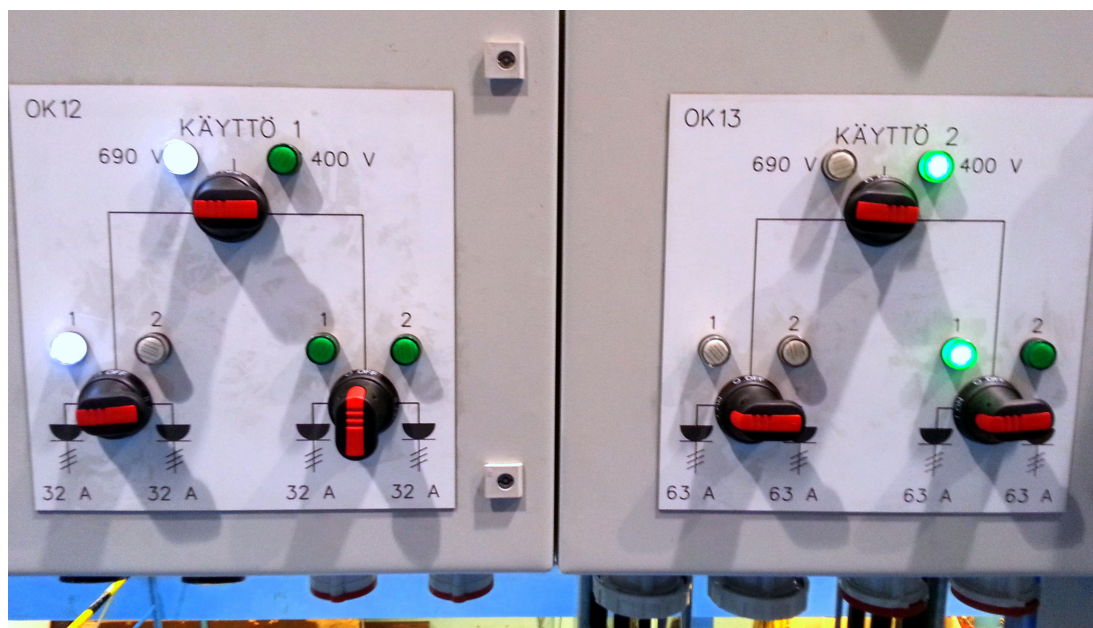
Lisäksi testattiin mahdollisia vikatilanteita ja esimerkiksi järjestelmän käyttäytymistä sähkökatkoksen jälkeen. Profinet-verkon vikaantumien testattiin irrottamalla verkko-kaapeli taajuusmuuttajasta ja katkaisemalla sähköt Profinet-verkon kytkimeltä. Moottorin käydessä, verkkokaapelin ”katkeaminen” aiheutti hälytyksen taajuusmuuttajaan ja moottori pysähtyi. Hätäseispiirin katkeaminen testattiin erillisellä testimuut-

tujalla. Todellisten kytkinten puuttuminen aiheutti myös hankaluuksia näissä testivaiheissa. Jännitteet ja käyttöpaikat testattiin ohjelmallisesti simuloimalla.

7.2 Käyttöönotto asiakkaan tiloissa

Käyttöönotto asiakkaan tiloissa aloitettiin tutustumalla kohteen ympäristöön ja koeajopaikan toimintaan. Ensimmäisenä käyttöliittymä esiteltiin uuden järjestelmän tuleville käyttäjille. Käyttöliittymä hyväksyttiin sellaisenaan, joten tässä vaiheessa visuaalisia tai toiminnallisia muutoksia ei lähdetty tekemään.

Ensimmäisessä testivaiheessa loput puuttuvat parametrit asetettiin taajuusmuuttajiin, tarkastettiin verkkoasetukset ja käyttöjen nimet. Lisäksi käyttöjen 2 ja 3 muuttajat tuli yhdistää ohjelmaan ja tarkastaa niiden toiminta, koska aiemmassa testivaiheessa logiikkaohjelma ja käyttöliittymä testattiin vain ensimmäisen käytön avulla lähes valmiiksi. Tässä vaiheessa jokaisen taajuusmuuttajan ohjaus viimeisteltiin identtiseksi, jotta jatkossa jokaisen osa-alueen testaus voidaan suorittaa heti kaikille taajuusmuuttajille järjestyksessä. Myös ohjelmamuutosten tekeminen helpottuu, koska muutos voidaan toteuttaa kaikkiin taajuusmuuttajiin samanaikaisesti. Seuraavaksi testattiin fyysisten jännitteenvälitsinkytken toiminta (Kuva 23). Tilatiedot tulivat logiikkaan niin kuin piti, lukuun ottamatta kahta kytkintä. Kytkinten apukärkien vaihdon jälkeen, kytkimet toimivat oikein ja näitä oli mahdollista käyttää testauksessa. Tilatietojen oikeellisuuden tarkastamisen jälkeen myös taajuusmuuttajien lukitusbitit testattiin valitsemalla tahallisesti väärät jännitteet kytkimillä.



Kuva 23. Käyttöjen 1 ja 2 moottorilähdöt sekä jännitteen ja käyttöpaikan valinnan keskukset

Seuraavaksi testattiin parametrien siirtyminen ja parametrikyselyn toiminta jokaisessa taajuusmuuttajassa. Parametrien kirjoitus ja luku onnistui hyvin, lukuun ottamatta virhettä koodissa, jonka johdosta parametrikysely lähetettiin aina vain samaan taajuusmuuttajaan. Tämän ongelman lisäksi parametrit pyöristyivät aina seuraavaan kokonaislukuun taajuusmuuttajaan lähetettäessä. Desimaalien pyöristyi tapahtui muunneltaessa moottoriparametrien arvoa REAL-tietotyyppistä UINT-tietotyyppiin. Tässä muunnoksessa desimaaliluvusta poistuvat pilkun jälkeisen numeron. Tämän olisi ollut mahdollista korjata vaihtamalla syklisessä tiedonsiirrossa siirrettävien datasanon tyyppiä. Esimerkiksi virtaparametri olisi pitänyt vaihtaa 16-bittisestä integer-tyypistä 32-bittiseen ”floating point”-datatyyppiin. Tämän lisäksi 32-bittisille datasanoille olisi pitänyt varata kaksi 16-bittistä datasanaa syklisestä kommunikoinnista. Tämä olisi puolestaan tuonut lisämuutoksia ohjelmaan ja täyttänyt PPO-tyypin mahdollistamien datasanon paikat. Ajanpuutteen vuoksi muutoksia ei kuitenkaan lähdetty tekemään.

Viimeisenä testattiin erikoistilanteet. Järjestelmän tuli toimia oikein hätä-seis-piiriin tai Profinet-verkon katkeamisesta ja taajuusmuuttajan vikaantumisesta ilmenneestä häiriöstä. Järjestelmän oikein toiminnalla tarkoitetaan käytössä olevan moottorin pysähtymistä vikatilanteesta sekä järjestelmän palautumista toimivaan tilaan häiriön poistuessa. Lisäksi järjestelmän tuli palautua sähkökatkosta samaan tilaan, kuin mis-

sä se oli ennen katkoa. Hätä-seis-piirin integrointi olemassa olevaan hätä-seis-järjestelmään rajattiin kuitenkin pois opinnäytetyöstä, joten hätäseispiirin tietoa simuloitiin edelleen testimuuttujalla.

Automaation testausaika asiakkaan tiloissa jäi vain noin kahteen päivään, koska laite-toimitukset myöhästyivät viikoilla. Asiakkaan tiloissa ei ollut mahdollista koestaa erillisillä irtomoottoreilla, ja hätä-seis-piirin muutokset toiseen järjestelmään olivat kesken, joten lopullinen laitteiston ajaminen oikeilla testimoottoreilla jätettiin aika-tilausyistä pois tästä insinöörityöstä. Näistä seikoista huolimatta, järjestelmän taajuusmuuttajia oli mahdollista koeajaa ilman moottoreita ja todeta niiden toimivuus. Kaksi taajuusmuuttajaa todettiin toimiviksi järjestelmässä. Käyttöliittymän oloarvojen testaus oli kuitenkin jätettävä viimeiseen testaukseen, koska ilman moottoreita ei ole saatavilla kunnollisia oloarvoja. Kolmatta taajuusmuuttajaa ei saatu pyörimään, koska käyntilupasignaalia ei jostain syystä saatu logiikkaan. Käyntilupasignaali pakotettiin taajuusmuuttajan ohjauspaneelistä kokoajan päällä olevaksi, mutta signaalia ei silti saatu näkyviin logiikassa tai väylässä. Kaikkien kolmen taajuusmuuttajan parametreja verrattiin, mutta vikaa ei löytynyt. Ainoa ero taajuusmuuttajilla oli ohjelmistojen versiot. Kolmannessa laitteessa oli vanhempi ohjelmistoversio kuin toimivissa laitteissa. Joissain tilanteissa myös versioristiriita saattaa aiheuttaa ongelmia. Asiasta oltiin yhteydessä ABB:n henkilöstöön, mutta se jäi kuitenkin vielä selvitettäväksi lopulliseen käyttöönottoon.

8 POHDINTA

8.1 Yhteenveto

Tämän insinöörityön päätavoitteet olivat rakennettavan järjestelmän logiikkaohjelman toteutus ja käyttöliittymän kehittäminen. Projektin päätavoitteet täyttyivät, koska lopputuloksena saatiin logiikkaohjelma ja moottoreiden koekäyttöjärjestelmän käyttöliittymä.

Projektin toteutumista arvioitiin lopuksi vertailemalla tuloksia logiikkaohjelmalle ja käyttöliittymälle asetettuihin vaatimuksiin. Taajuusmuuttajien käynnistäminen ja pysäyttäminen toteutettiin ohjelmallisesti niin kuin alun perin suunniteltiin. Myös käyttäjän valitsema jännite siirretään automaattisesti taajuusmuuttajiin, jolloin käyttäjällä on mahdollisuus vaikuttaa käytettävään jännitteeseen vain fyysisten kytkinten avulla. Asetettujen vaatimusten mukaan hätä-seis-toiminnot tuli integroida olemassa olevaan järjestelmään. Hätä-seis-piirin fyysinen liittäminen järjestelmään rajattiin pois tästä opinnäytetyöstä, mutta logiikkaohjelma toteutettiin myös tämän vaatimuksen mukaisesti. Alkuperäisen suunnitelman mukaan liian suurten tai pienten moottoriparametrien käyttäminen tuli estää logiikkaohjelmassa. Raja-arvojen valvonta kehitettiin ja todettiin toimivaksi, mutta valvonta toteutettiin logiikkaohjelman sijasta käyttöliittymään. Logiikkaohjelmaa on myös mahdollista muuttaa helposti niin, että raja-arvoja ja käyttäjän antamia arvoja voidaan valvoa vertailemalla niitä keskenään.

Logiikkaohjelman lisäksi vaatimuksiin päästiin myös käyttöliittymän kehittämisessä. Lopullisesta käyttöliittymästä saatiin visuaalisesti helppokäyttöinen ja ymmärrettävä, vaikka asiakkaalle ei ollut selvää näkemystä siitä, millainen käyttöliittymän tulisi olla. Käyttöönottovaiheen alkaessa käyttöliittymä hyväksytettiin sen tulevalle käyttäjällä. Käyttöliittymä hyväksyttiin sellaisenaan. Toisaalta, lopullinen mielipide käyttöliittymän toiminnoista ja ominaisuuksista saadaan vasta muutaman viikon työskenteilyn jälkeen. Käyttöliittymän etusivulle lisättiin vaatimuksissa määriteltyjen moottorin ottaman virran ja tehon oloarvojen lisäksi muita oleellisia tietoja taajuusmuuttajalta. Myös muihin käyttöliittymää koskeviin vaatimuksiin päästiin pienistä ongelmista huolimatta.

Tavoitteisiin päästiin myös testaus ja käyttöönottovaiheessa, lukuun ottamatta viimeistä käyttöönottoa testimoottoreiden avulla. Laitetoimituksista johtuneen projektin viivästymisen takia automaatiolle suunniteltua testausaikaa toimeksiantajan ja asiakkaan tiloissa jouduttiin lyhentämään. Alkuperäisen suunnitelman mukaan automaation testaamiseen käytettävissä oleva aika toimeksiantajan tiloissa oli noin viikko ja asiakkaan tiloissa 1 - 2 viikkoa. Mekaanisesti valmista järjestelmää päästiin testaamaan kuitenkin vasta kaksi päivää ennen, kuin järjestelmä tuli siirtää asiakkaan tiloihin asennettavaksi. Lisäksi asennusvaiheessa toimitettu väärän kaapeli aiheutti kolmen päivän viivästymisen, jolloin käyttöönottoon jäi aikaa vain kaksi päivää. Näistä

seikoista huolimatta järjestelmä testattiin asiakkaan tiloissa lähes käyttövalmiiksi opinnäytetyöprojektin lopuksi. Lopullinen käyttöönotto ja käytön aikana eteen tulevat tarpeelliset ohjelma- ja käyttöliittymämuutokset rajattiin pois tästä insinööritoiminnasta aikataulusyistä.

Kokonaisuudessaan projekti eteni hyvin. Pää tavoitteet täyttyivät ja järjestelmä ehdittiin testata asiakkaan tiloissa lähes käyttövalmiiksi. Projekti eteni ilman suurempia ongelmia. Eteen tulleet ongelmat liittyivät suurimmaksi osaksi uusien ohjelmien käyttöön. Näistä ongelmista päästiin yli kuitenkin melko helposti.

8.2 Kehitysideoita

Todelliset logiikkaohjelman ja käyttöliittymän kehitystarpeet selviävät vasta, kun järjestelmä on ollut käytössä jonkin aikaa. Yksi käyttöliittymän kehitysmahdollisuuksista voisi olla erillisen tietokannan lisääminen koekäytettäville moottoreille. Tietokantaan määriteltäisiin EXCEL-tilukon muodossa ne moottorit, joita useimmiten käytetään. Käyttöliittymän käyttökohtaisille sivuille puolestaan lisättäisiin käyttäjälle mahdollisuus valita tietokannasta käytettävä moottori, tai syöttää arvot käsin. Parametrien syöttäminen käsin voitaisiin mahdollisesti jopa kokonaan korvata tietokannalla, jos esimerkiksi huoltosivulle määriteltäisiin toiminnot, joilla olisi mahdollista lisätä ja poistaa moottoreita tietokannasta. Moottorit nimettäisiin yksilöllisesti ja esimerkiksi kilpiarvot voitaisiin näyttää moottoria valitessa. Tämä nopeuttaisi työntekoa ja käyttäjän tekemät näppäilyvirheet vähenisivät. Huoltosivu olisi mahdollista myös suojata salasanalla, jolloin vain tietyillä henkilöillä olisi pääsy moottoritietokantaan.

8.3 Oppimistulokset

Työ oli erittäin mielenkiintoinen, koska uuden järjestelmän toteuttamiselle ei ollut paljonkaan vaatimuksia, joten ohjelmakoodin ja käyttöliittymän visuaalisen ilmeen toimintoinen sai kehittää alusta loppuun itse. Myös opintojen kannalta projekti oli opettavainen ja projektista on varmasti hyötyä tulevassa työssäni automaatioprojektien parissa.

Tavoitteeni oli pitää opinnäytetyöraportti mahdollisimman yksinkertaisena ja selkeänä, vaikka uusien asioiden ymmärtämiseksi jouduin käymään läpi paljon teoriaa ja tekemään havainnollistavia kuvia. Pyrin myös käyttämään mahdollisimman uusia ja luotettavia lähteitä. Työn tekemistä hankaloitti jatkuva projektin viivästyminen, joka johtui pitkittyneistä laitetoimitusajoista ja vääristä kaapelitoimituksista. Toisaalta, pitkittyneet laitetoimitukset antoivat lisää aikaa perehtyä Profinet-tekniikkaan ja uusiin ohjelmistoihin.

Myös valtava materiaalin määrä toi omat haasteensa esimerkiksi Profinet-materiaalin luotettavuuden analysoinnissa. Suurin haaste tässä projektissa kuitenkin oli laajojen ohjelmistojen sujuvan käytön opettelu. Jos ohjelmat olisivat olleet ennestään tuttuja, ohjelmointi olisi sujunut merkittävästi nopeammin. Näistä seikoista huolimatta ABB:n taajuusmuuttajat, Profinet-tekniikka, Control Builder ja Web Studio tulivat tutuiksi. Varsinkin juuri käytännön työskentely opetti paljon uutta ja moni asia selvisi vasta laitteen manuaalia lukemalla sekä kokeilemalla eri ratkaisuja käytännössä.

Kaiken kaikkiaan, vaikka kaikki oppiminen ei näy tekstistä, projekti oli erittäin opettavainen ja selkeytti projektitoimintaa yleiseltä tasolta aina käytännön tekemisen ja teoriaan syventymisen muodossa.

LÄHTEET

ABB Group. 2014. Frequency Converter. Viitattu 18.2.2014. <http://www.abb.com>.

ABB Oy. 2013. ABB Industrial drives. Firmware manual ACS880 primary control program. Viitattu 13.1.2014.
[http://www05.abb.com/global/scot/scot201.nsf/veritydisplay/51fff3d32ef8b701c1257ba6004d0e34/\\$file/EN_ACS880_FW_Man_G_A4.pdf](http://www05.abb.com/global/scot/scot201.nsf/veritydisplay/51fff3d32ef8b701c1257ba6004d0e34/$file/EN_ACS880_FW_Man_G_A4.pdf)

ABB Oy. 2011. ABB Industrial drives. User's manual FENA-01/-11 Ethernet adapter module. Viitattu 13.1.2014.
[http://www05.abb.com/global/scot/scot201.nsf/veritydisplay/74152fb1ab82ab43c125790a00496499/\\$file/EN_FENA01_11_UM_A_screenres.pdf](http://www05.abb.com/global/scot/scot201.nsf/veritydisplay/74152fb1ab82ab43c125790a00496499/$file/EN_FENA01_11_UM_A_screenres.pdf)

Asmala, H. 2013. Ohjelmistoarkkitehtuuri. Kurssimateriaali.

Asmala, H., Koskinen, K., Koskela, M., Mätäsniemi, T., Soini, A., Strömman, M., Tommila, T., Valkonen, J. n.d. Automaatiosovellusten ohjelmistokehitys: Suunnittelun työtavat, välineet ja sovellusarkkitehtuurit. Helsinki: Suomen Automaatioseura ry

Bernhard, H. & Mottok, J. 2014. Real-time behaviour of Ethernet on the example of Profinet. Viitattu 15.2.2014. http://www.hs-regensburg.de/fileadmin/media/fakultaeten/ei/forschung_projekte/MAPR_Ver%C3%B6ffentlichungen/ARC_Heitzer.pdf

Control Builder (Version 2.3). 2013. ABB OY.

General Electric. 2014. AF-650 GP & AF-600 FP Profinet – Operating Instructions. Viitattu 23.4.2014. <http://apps.geindustrial.com/publibrary/checkout/DET-744?TNR=Installation%20and%20Instruction|DET-744|generic>

GE Intelligent Platforms. 2013. Profinet for Network Geeks. Viitattu 28.1.2014. <http://www.ge-ip.com/blog/profinet-network-geeks-want/>

Gory, E. 2009. PROFINET IO Basic. Profinet International. Viitattu 17.3.2014. <http://www.profibus.ie/PROFINET%20IO%20Basic.pdf>

Heimbürger, H., Markkanen, P., Norros, L., Paunonen, H., Savioja, P., Sundquist, M., Tommila, T. 2011. Valvomo: Suunnittelun periaatteet ja käytännöt. Helsinki: Suomen Automaatioseura ry.

IFAC. 2005. Measurements of telegram-sequences with Ethereal. Viitattu 17.3.2014. http://www.felser.ch/download/6_PN_Measurements_E02.pdf

Jaakonhuhta, H. 2005. Lähiverkot – Ethernet. Helsinki: EDITA.

PI North America. 2011. PROFIBUS Nutzerorganisation. Profinet System Description – Technology and Application. Viitattu 13.1.2014. http://us.profinet.com/wp-content/uploads/2012/11/PI_PROFINET_System_Description_EN_web.pdf

PI North America. 2014. PROFINET – Industrial Ethernet for advanced manufacturing. Viitattu 11.2.2013. <http://us.profinet.com/technology/profinet/>

Sataservice Oy:n www-sivut. 2012. Viitattu 10.2.2014. <http://www.sataservice.fi>

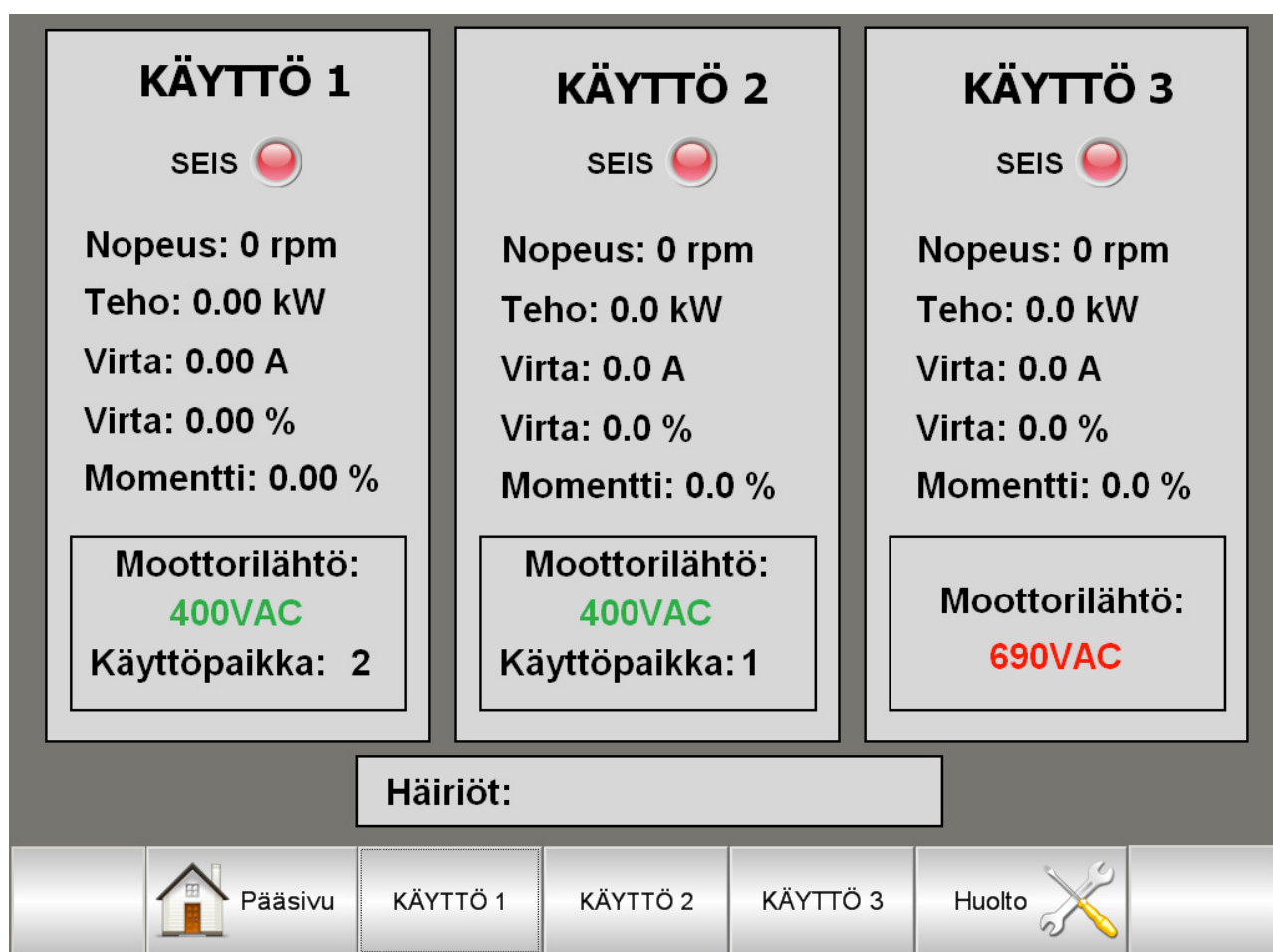
Siemens AG. 2008. Drive Technologies and Industry Automation. Real Time on Industrial Ethernet. Viitattu 22.4.2014. http://www.siemens.fi/pool/products/industry/iadt_is/tuotteet/automaatiotekniikka/teollinen_tiedonsiirto/profinet/br_ertec.pdf

Suvela, T. 2011. Teollisuus-Ethernet. Viitattu 20.2.2014.

Tuomala, M. Laitevalinnat. Vastaanottaja: joni.liukkonen@student.samk.fi. Lähetetty 19.3.2014. Viitattu 24.3.2014.

Web Studio (Version 1.7). 2013. Indusoft.

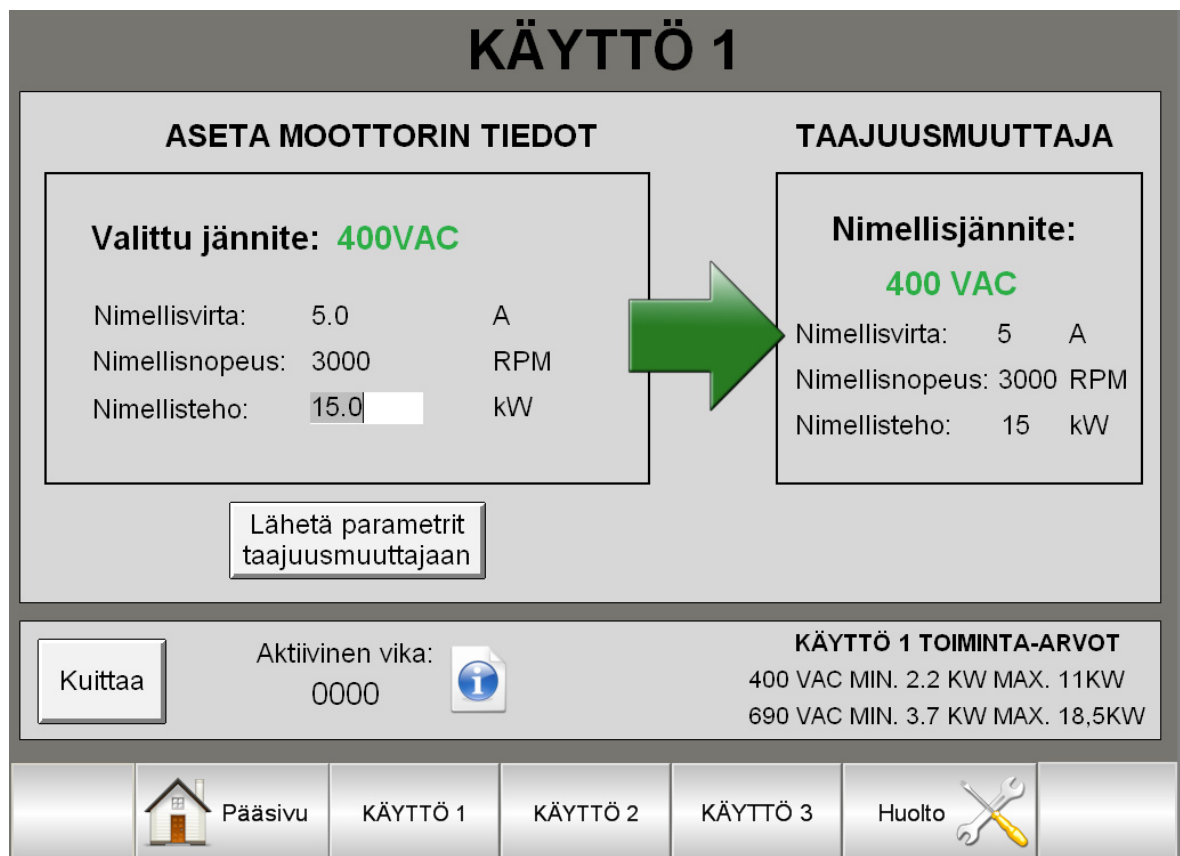
KÄYTTÖLIITTYMÄ



Kuva 1 Käyttöliittymän pääsivu, kun käytöt ovat Seis-tilassa.



Kuva 2 Käyttöliittymän pääsivu häiriötilanteessa



Kuva 3 Käyttökohtainen sivu normaalissa toimintatilassa

KÄYTTÖ 3

ASETA MOOTTORIN TIEDOT

Valittu jännite: 690VAC

Nimellisvirta: 81.0 A

Nimellisaika: 1480 RPM

Nimellisteho: 45.0 kW

➔

TAAJUUSMUUTTAJA

Nimellisjännite: 690 VAC

Nimellisvirta: 81 A

Nimellisaika: 1480 RPM

Nimellisteho: 45 kW

Lähetä parametrit taajuusmuuttajaan

Kuittaa

Häiriö!

Aktiivinen vika:

7510

KÄYTTÖ 3 TOIMINTA-ARVOT

400 VAC MIN. 18 kW MAX. 90 kW

690 VAC MIN. 32 kW MAX. 160 kW

Pääsivu

KÄYTTÖ 1

KÄYTTÖ 2

KÄYTTÖ 3

Huolto

Kuva 4 Käyttökohtainen sivu taajuusmuuttajan häiriötilanteessa

KÄYTTÖ 1

Häiriö!
SEIS
Häiriö!

Nopeus: 0 rpm

Teho: 0.00 kW

Virta: 0.00 A

Virta: 0.00 %

Momentti: 0.00 %

Moottorilähtö:

400VAC

Käyttöpaikka: 2

KÄYTTÖ 2

Häiriö!
SEIS
Häiriö!

Nopeus: 0 rpm

Teho: 0.0 kW

Virta: 0.0 A

Virta: 0.0 %

Momentti: 0.0 %

Moottorilähtö:

400VAC

Käyttöpaikka: 1

KÄYTTÖ 3

Häiriö!
SEIS
Häiriö!

Nopeus: 0 rpm

Teho: 0.0 kW

Virta: 0.0 A

Virta: 0.0 %

Momentti: 0.0 %

Moottorilähtö:

690VAC

Kuva 5 Käyttöliittymän pääsivu, kun kaikissa taajuusmuuttaissa häiriötila päällä